

**INTEGRASI DATA DAN VISUALISASI
MENGUNAKAN METODE *TREEMAP*
DI PT. CHEVRON PACIFIC INDONESIA**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

ADITYA AYU HAPSARI
10951005553



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
2013**

LEMBAR PERSETUJUAN

**INTEGRASI DATA DAN VISUALISASI
MENGUNAKAN METODE *TREEMAP*
DI PT. CHEVRON PACIFIC INDONESIA**

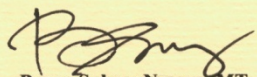
TUGAS AKHIR

oleh:

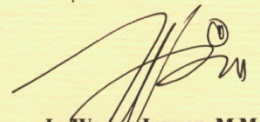
ADITYA AYU HAPSARI
10951005553

Telah diperiksa dan disetujui sebagai laporan tugas akhir
di Pekanbaru, pada tanggal 29 Oktober 2013

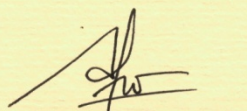
Pembimbing I


Benny Sukma Negara, MT
NIP. 19820313 200901 1 009

Pembimbing II


Ir. Wayan Irawan, M.M

Koordinator Tugas Akhir


Muhammad Affandes, MT
NIK. 130510030

LEMBAR PENGESAHAN
INTEGRASI DATA DAN VISUALISASI
MENGGUNAKAN METODE *TREEMAP*
DI PT. CHEVRON PACIFIC INDONESIA

TUGAS AKHIR

Oleh :

ADITYA AYU HAPSARI
10951005553

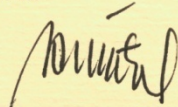
Telah dipertahankan di depan sidang dewan penguji
Sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika
Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau
Di Pekanbaru, pada tanggal, 29 Oktober 2013

Pekanbaru, 29 Oktober 2013

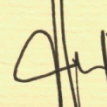
Mengesahkan,

Ketua Jurusan

Dekan



Dra. Hj. Yenita Morena, M.Si
NIP. 19601125 198503 2 002



Elin Haerani, ST, M.Kom
NIP. 19810513 200710 2 003

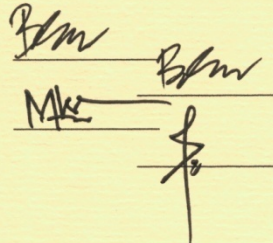
DEWAN PENGUJI

Ketua : Benny Sukma Negara, M.T

Sekretaris : Benny Sukma Negara, MT

Anggota I : Muhammad Irsyad, MT

Anggota II : Iwan Iskandar, MT



LEMBAR PENGESAHAN PERUSAHAAN
VISUALISASI DAN PERENCANAAN INTEGRASI
DATA MENGGUNAKAN METODE TREEMAP
DI PT.CHEVRON PACIFIC INDONESIA

LAPORAN TUGAS AKHIR

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau


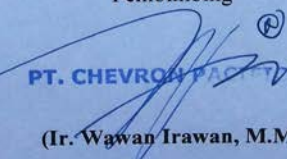

Oleh:

ADITYA AYU HAPSARI

10951005553

Telah disetujui dan disahkan sebagai laporan Tugas Akhir
di Pekanbaru, pada tanggal

Pekanbaru, 04 Juni 2013
Pembimbing

  
(Ir. Wawan Irawan, M.M)

LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL

Tugas Akhir yang tidak diterbitkan ini terdaftar dan tersedia di Perpustakaan Universitas Islam Negeri Sultan Syarif Kasim Riau adalah terbuka untuk umum dengan ketentuan bahwa hak cipta pada Penulis. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau ringkasan hanya dapat dilakukan seizin Penulis dan harus disertai dengan kebiasaan ilmiah untuk menyebutkan sumbernya.

Penggandaan atau penerbitan sebagian atau seluruh Tugas Akhir ini harus memperoleh izin dari Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau. Perpustakaan yang meminjamkan Tugas Akhir ini untuk anggotanya diharapkan untuk mengisi nama, tanda peminjaman dan tanggal pinjam.

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan didalam daftar pustaka.

Pekanbaru, 29 Oktober 2013

Yang membuat pernyataan,

ADITYA AYU HAPSARI

LEMBAR PERSEMBAHAN

Karya ini kupersembahkan untuk:

- ❖ Allah SWT Yang Maha Pengasih dan Maha Penyayang, yang telah memberikan segalanya. Syukur atas nikmat dan karunia yang diberikan-NYA.
- ❖ Kedua orang tuaku tercinta, Bapak Kiesworo Hartono dan Ibu Sumarti yang setiap ucapannya adalah doa dan langkahnya adalah dorongan sehingga Penulis dapat menyelesaikan Tugas Akhir ini.
- ❖ Mas Harma dan Mas Tetra yang selalu membantu dan memberikan motivasinya.

INTEGRASI DATA DAN VISUALISASI MENGUNAKAN METODE *TREEMAP* DI PT. CHEVRON PACIFIC INDONESIA

ADITYA AYU HAPSARI
10951005553

Tanggal sidang : 29 Oktober 2013

Periode wisuda : November 2013

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas KM 15 No. 155 Pekanbaru

ABSTRAK

PT. Chevron Pacific Indonesia (PT.CPI) merupakan perusahaan energi terbesar yang bergerak di bidang perminyakan dan gas. Sebagai perusahaan besar PT.CPI memerlukan adanya pendataan mengenai setiap kegiatan eksplorasi dan produksi yang dihasilkan, seperti data minyak, gas, air, *steam*, *fluid*, LPO (*Loss Production Oil*), status *well*, data eksplorasi dan produksi lainnya yang ada pada setiap *well* di seluruh area dan daerah operasi PT. Chevron Pacific Indonesia. Data hasil eksplorasi dan produksi yang tersedia akan diproses dan dikalkulasikan sesuai kriteria datanya. Dengan banyaknya data yang berasal dari berbagi sumber data diperlukan proses untuk mengintegrasikannya dalam sebuah kesatuan data. Agar hasil data yang telah terintegrasi dapat mudah dilihat, dipahami, dan cepat dalam mengambil keputusan maka dibutuhkan sebuah sistem yang dapat memvisualisasikannya dalam bentuk peta data berdasarkan kebutuhan setiap *user* (*manager* dan pimpinan tim). Sistem ini dibangun menggunakan metode *Treemap*. Metode ini sangat sesuai karena dapat memvisualisasikan data berdasarkan *group* data dan kategori warna dengan menyeluruh dan detail untuk setiap kondisi data *well* yang terjadi.

Kata Kunci: Data Eksplorasi dan Produksi, Integrasi Data, *Treemap* dan Visualisasi.

INTEGRATION DATA AND VISUALIZATION USING TREEMAP METHOD AT PT. CHEVRON PACIFIC INDONESIA

ADITYA AYU HAPSARI
10951005553

Date of Final Exam : October 29th, 2013

Graduation Ceremony Period : November 2013

Informatics Engineering Departement
Faculty of Sciences and Technology
State Islamic University of Sultan Syarif Kasim Riau
Jl. Soebrantas KM 15 No. 155 Pekanbaru

ABSTRACT

PT. Chevron Pacific Indonesia (PT.CPI) is the biggest energy company specially in oil and gas. Because of its complexity, PT. CPI need collection and manage of any exploration and production activities data is about oil data, gas, water, steam, fluid, LPO (Loss Production Oil), well status, other data that in each well throughout the area and operation area of PT. Chevron Pacific Indonesia. The results of exploration and production data is available will be processed and calculated according to criteria. Because of the number of data its needed to integrate processes in a unified data. The integrated data can be easily seen, understanding, and quick in taking the decision its needed a system that can visualize the data in the form of maps based on each users (manager and team leader) purpose. The system is built by treemap. This method is very suitable because it can visualize the data based on group data and color-coded categories with thorough and detail to every condition that occurs the data well .

Keywords : Exploration and Production Data, Integration Data and Treemap Visualization.

KATA PENGANTAR



Assalamu'alaikum W.W.

Puji syukur penulis panjatkan kehadiran Allah SWT, yang telah memberikan rahmat dan hidayah-Nya, sehingga Penulis dapat menyelesaikan laporan Tugas Akhir ini dengan sebaik-baiknya.

Laporan Tugas Akhir yang berjudul “Integrasi Data dan Visualisasi Menggunakan Metode *Treemap* di PT. Chevron Pacific Indonesia” ini disusun dan ditulis dalam rangka melaksanakan kurikulum Program Studi Teknik Informatika dan mengenal serta memahami penerapan Ilmu Teknik Informatika di lapangan. Pada kesempatan ini penulis menghaturkan terima kasih yang sebesar-besarnya kepada :

1. ALLAH SWT yang telah memberikan limpahan rahmat dan hidayah-Nya.
2. Rasulullah SAW, seorang pimpinan umat islam sekaligus panutan umat islam yang tiada henti-hentinya berjuang demi menyebarkan agama Islam.
3. Ayahanda Kiesworo Hartono, Ibunda Sumarti tercinta yang telah memberikan dukungan baik moril maupun materiil. Terima kasih, tanpa Ayahanda dan Ibunda, Penulis bukan apa-apa. Cinta dan kasih sayang dari Ayahanda dan Bunda akan selalu kuingat selamanya.
4. Mas Tetra Hapsoro dan Mas Harma Didy Wicaksono selaku mas Penulis yang telah banyak memberi bantuan, kritikan, nasehat dan banyak lagi. Engkau telah banyak memberi pengalaman dan arti kehidupan yang berharga dalam hidup Penulis.
5. Bapak Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
6. Ibu Dra. Hj. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
7. Bapak Muhammad Irsyad, MT, selaku Kepala Jurusan Teknik Informatika dan selaku penguji I Tugas Akhir.

8. Ibu Elin Haerani, ST, M.Kom, selaku sekretaris Jurusan Teknik Informatika.
9. Bapak Muhammad Affandes, MT, selaku koordinator Tugas Akhir.
10. Bapak Benny Sukma Negara, MT, selaku Dosen Pembimbing I yang telah memberikan bimbingan dan pengarahan kepada Penulis dalam menyelesaikan Tugas Akhir ini.
11. Bapak Ir. Wawan Irawan, M.M, selaku Pembimbing II di PT.Chevron Pacific Indonesia, yang telah meluangkan waktunya untuk memberikan pengarahan dan bimbingan kepada Penulis dalam menyelesaikan Tugas Akhir ini.
12. Bapak Iwan Iskandar, MT, selaku penguji II Tugas Akhir.
13. Bapak Elwin F.Nasution selaku *Manager* HR-Rumbai PT. Chevron Pacific Indonesia yang telah menempatkan Penulis melakukan penelitian.
14. Teman-teman seperjuangan Penulis di kampus UIN Suska Riau, TIF B '09, terima kasih atas dukungan dan doanya.
15. Semua pihak yang telah memberikan bantuannya dari awal hingga akhir laporan Tugas Akhir Penulis selesai, yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa penyusunan dan penulisan laporan Tugas Akhir ini masih jauh dari kesempurnaan. Untuk itu, Penulis mengharapkan saran dan kritik yang bersifat membangun demi peningkatan kemampuan penulis di masa yang akan datang.

Akhir kata, semoga laporan Tugas Akhir ini dapat memberikan manfaat dan sumbangan bagi kita semua. Amin.

Wassalammu'alaikum Wr.Wb

Pekanbaru, 29 Oktober 2013
Penulis

Aditya Ayu Hapsari

DAFTAR ISI

LEMBAR PERSETUJUAN	II
LEMBAR PENGESAHAN.....	III
LEMBAR PENGESAHAN PERUSAHAAN	IV
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	IV
LEMBAR PERNYATAAN	VI
LEMBAR PERSEMBAHAN.....	VII
ABSTRAK	VIII
<i>ABSTRACT</i>	IX
KATA PENGANTAR	X
DAFTAR ISI	XII
DAFTAR GAMBAR	XVI
DAFTAR TABEL	XVIII
DAFTAR ISTILAH.....	XXI
DAFTAR SIMBOL	XXII
BAB I PENDAHULUAN	I-1
1.1. Latar Belakang.....	I-1
1.2. Rumusan Masalah.....	I-3
1.3. Batasan Masalah	I-3
1.4. Tujuan Penelitian	I-3
1.5. Sistematika Penulisan.....	I-4
BAB II LANDASAN TEORI	II-1
2.1 Model <i>Waterfall</i>	II-1
2.2 Integrasi Data.....	II-2
2.3 Sistem Basis Data.....	II-3
2.3.1. Penyusunan Sistem Basis Data.....	II-4
2.3.2. Tujuan Basis Data.....	II-5
2.3.3. Komponen-komponen Sistem Basis Data.....	II-7
2.4 Oracle	II-8
2.4.1. Oracle dan ORDBMS	II-8
2.4.2. Kehandalan <i>Database</i> Oracle.....	II-8

2.4.3. SQL	II-9
2.4.4. PL/SQL	II-10
2.5 <i>Unified Modeling Language (UML)</i>	II-10
2.5.1. Tujuan <i>Unified Modelling Language (UML)</i>	II-11
2.5.2. Konsep Dasar UML	II-11
2.6 Java.....	II-13
2.7 <i>Treemap</i>	II-16
2.7.1. Sejarah <i>Treemap</i>	II-16
2.7.2. Metode <i>Treemap</i>	II-17
2.8 VBScript	II-20
2.9 Visualisasi.....	II-21
2.10 Pemrograman Web.....	II-22
2.10.1. HTML	II-23
2.10.2. PHP	II-24
BAB III METODOLOGI PENELITIAN.....	III-1
3.1. Alur Metodologi Penelitian	III-1
3.2. Pengumpulan Data	III-2
3.3. Analisa.....	III-3
3.4. Perancangan	III-3
3.5. Implementasi dan Pengujian.....	III-3
3.5.1. Implementasi	III-3
3.5.2. Pengujian.....	III-4
3.6. Kesimpulan dan Saran.....	III-4
BAB IV ANALISA DAN PERANCANGAN	IV-1
4.1 Analisa Sistem	IV-1
4.1.1. Subjek Penelitian	IV-1
4.1.2. Analisa Sistem Lama	IV-1
4.1.3. Analisa Sistem Baru.....	IV-3
4.1.4. Analisa Data	IV-3
4.1.5. Integrasi Data	IV-3
4.1.6. Analisa Kebutuhan Data	IV-5
4.1.6.1. Dsc Last Well Data Wo Shutin	IV-5

4.1.6.2.	KLO Asset Doc	IV-10
4.1.6.3.	KLO Asset Com	IV-10
4.1.7.	Metode <i>Treemap</i>	IV-11
4.1.8.	<i>Treemap Architecture Diagram</i>	IV-13
4.1.9.	<i>Typical Data Flow Treemap</i>	IV-14
4.1.10.	Spesifikasi <i>Software</i> dan <i>Hardware</i> yang di Butuhkan	IV-16
4.1.10.1.	Minimal <i>Requirement</i>	IV-16
4.1.10.2.	Perancangan Lingkungan Pengembang	IV-17
4.1.11.	<i>Unified Modeling Language</i> (UML).....	IV-17
4.1.11.1.	<i>Use Case Diagram</i>	IV-17
4.1.11.2.	<i>Sequence Diagram</i>	IV-19
4.1.11.2.1.	<i>Sequence Diagram</i> Mengelola Data	IV-19
4.1.11.2.2.	<i>Sequence Diagram</i> Visualisasi <i>Treemap</i>	IV-20
4.1.11.2.3.	<i>Sequence Diagram</i> Menjalankan Laporan	IV-21
4.1.11.3.	<i>Class Diagram</i>	IV-21
4.1.11.4.	<i>Activity Diagram</i>	IV-22
4.1.11.4.1.	<i>Activity Diagram</i> Mengelola Data	IV-22
4.1.11.4.2.	<i>Activity Diagram</i> Visualisasi <i>Treemap</i>	IV-22
4.1.11.4.3.	<i>Activity Diagram</i> Menjalankan Laporan	IV-23
4.2	Perancangan Perangkat Lunak.....	IV-24
4.2.1.	Tipe Data Oracle Yang Digunakan.....	IV-24
4.2.2.	Entity Relationship Model (ERM)	IV-24
4.2.3.	Kamus Data (<i>Data Dictionary</i>)	IV-27
4.2.4.	Perancangan Tabel.....	IV-28
4.2.5.	Perancangan <i>View</i>	IV-31
4.2.6.	Perancangan <i>Procedure</i>	IV-34
4.2.7.	Perancangan Antar Muka.....	IV-35
4.2.7.1.	<i>Home</i>	IV-36
4.2.7.2.	<i>Treemap</i> KLO	IV-36
4.2.7.3.	<i>Report</i>	IV-37
BAB V IMPLEMENTASI DAN PENGUJIAN.....		V-1

5.1. Implementasi (<i>Implementation</i>)	V-1
5.1.1. Batasan Implementasi	V-1
5.1.2. Lingkungan Implementasi.....	V-1
5.1.3. Analisis Hasil	V-2
5.1.4. <i>Script Load Data</i>	V-2
5.1.4.1. <i>Script Load Data Otomatis</i>	V-2
5.1.4.2. <i>Script Load Data Manual</i>	V-4
5.1.5. Implementasi Tampilan Sistem	V-5
5.1.5.1. Tampilan Menu Utama	V-5
5.1.5.2. Tampilan Menu <i>Treemap</i> KLO.....	V-5
5.1.5.3. Tampilan Menu <i>Report</i>	V-6
5.2. Pengujian Sistem.....	V-7
5.3. Pengujian Sistem dengan <i>Black Box</i>	V-7
5.3.1. Modul Pengujian <i>Script Load Data</i>	V-8
5.3.2. Modul Pengujian Tampil <i>Treemap</i> KLO	V-9
5.3.3. Modul Pengujian Laporan.....	V-9
5.4. Kesimpulan Pengujian.....	V-10
BAB VI PENUTUP	VI-1
6.1. Kesimpulan	VI-1
6.2. Saran.....	VI-1
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar	Halaman
2. 1 Tahapan dalam <i>waterfalls</i>	II-1
2. 2 Keunggulan-keunggulan Java	II-14
2. 3 <i>Tree diagram</i> dan <i>Treemap</i>	II-19
3. 1 Alur metodologi penelitian.....	III-1
4. 1 <i>Flowchart</i> aktivitas sistem lama	IV-2
4. 2 <i>Flowchart</i> integrasi data.....	IV-4
4. 3 <i>KLO well exception tree model</i> – Dsc last well data wo shutin fase I.....	IV-6
4. 4 <i>KLO well exception tree model</i> – Dsc last well data wo shutin fase II.....	IV-7
4. 5 <i>KLO well exception tree model</i> – Dsc last well data wo shutin fase III ...	IV-7
4. 6 <i>KLO well exception tree model</i> – Dsc last well data wo shutin fase IV ...	IV-8
4. 7 <i>KLO well exception tree model</i> – KLO asset doc.....	IV-10
4. 8 <i>KLO well exception tree model</i> – KLO asset com.....	IV-10
4. 9 Level struktur <i>tree</i> dengan nomor indikasi ukuran dari setiap <i>leaf</i> <i>node</i>	IV-11
4. 10 Implementasi <i>Treemap</i>	IV-11
4. 11 <i>Treemap architecture diagram</i>	IV-13
4. 12 <i>Typical data flow treemap</i>	IV-14
4. 13 <i>Use case diagram</i>	IV-18
4. 14 <i>Sequence diagram</i> mengelola data	IV-20
4. 15 <i>Sequence diagram</i> visualisasi <i>Treemap</i>	IV-20
4. 16 <i>Sequence diagram</i> menjalankan laporan.....	IV-21
4. 17 <i>Class diagram</i>	IV-21
4. 18 <i>Activity diagram</i> mengelola data	IV-22
4. 19 <i>Activity diagram</i> visualisasi <i>Treemap</i>	IV-23
4. 20 <i>Activity diagram</i> menjalankan laporan.....	IV-23
4. 21 <i>Entity relationship model (ERM)</i>	IV-25
4. 22 Rancangan struktur menu	IV-35
4. 23 Perancangan menu <i>home</i>	IV-36
4. 24 Perancangan <i>Treemap</i> KLO	IV-36

4. 25 Perancangan menu <i>Report</i>	IV-37
5. 1 <i>Script load data fase I</i>	V-3
5. 2 <i>Script load data fase II</i>	V-3
5. 3 <i>Script load data fase III</i>	V-4
5. 4 Tampilan menjalankan <i>load script data manual</i>	V-4
5. 5 Tampilan menu utama.....	V-5
5. 6 Tampilan menu <i>Treemap KLO</i>	V-6
5. 7 Tampilan menu <i>report</i>	V-6
5. 8 <i>Report excel</i>	V-7

DAFTAR TABEL

Tabel	Halaman
4. 4 Spesifikasi <i>use case</i> mengelola data	IV-18
4. 5 Spesifikasi <i>use case</i> pemilihan <i>group</i> data.....	IV-18
4. 6 Spesifikasi <i>use case</i> Visualisasi <i>Treemap</i>	IV-19
4. 7 Spesifikasi <i>use case</i> laporan.....	IV-19
4. 8 Tipe data yang digunakan di Oracle	IV-24
4. 9 Keterangan ERM	IV-26
4. 10 Kamus data dsc last well data wo shutin.....	IV-27
4. 11 Kamus data dsc lpo day wo shutin	IV-27
4. 12 Basis data dsc last well data wo shutin	IV-28
4. 13 Basis data dsc lpo day wo shutin	IV-30
4. 14 View v dsc well except efficiency	IV-32
4. 15 View v dsc prodeff sum wo shutin	IV-33
A. 1 Kamus data KLO well prod day	A-1
A. 2 Kamus data KLO well prod day	A-6
A. 3 Kamus data KLO prod defer day.....	A-6
A. 4 Kamus data well last data.....	A-7
A. 5 Kamus data KLO well daily data	A-7
A. 6 Kamus data KLO well test	A-8
A. 7 Kamus data prod string	A-8
A. 8 Kamus data PPDM38 well daily data	A-9
A. 9 Kamus data PPDM38 pden flow measurement.....	A-9
A. 10 Kamus data PPDM38 well deferement daily	A-10
A. 11 Kamus data PPDM38 pden vol prod day.....	A-10
A. 12 Kamus data PPDM38 prod string alias	A-11
A. 13 Kamus data PPDM38 pden vol prod defer day	A-11
A. 14 Kamus data time dim	A-12
A. 15 Kamus data string status hist	A-12
A. 16 Kamus data PPDM38 prod str stat hist	A-13
A. 17 Kamus data well job	A-13

A. 18 Kamus data PPDM38 work order.....	A-14
A. 19 Kamus data PPDM38 work order component.....	A-14
A. 20 Kamus data PPDM38 vt prod string	A-15
A. 21 Kamus data KLO asset doc	A-15
A. 22 Kamus data KLO asset com.....	A-16
B. 1 Basis data well prod day	B-1
B. 2 Basis data prod defer day	B-2
B. 3 Basis data well last data	B-3
B. 4 Basis data well daily data.....	B-5
B. 5 Basis data well test.....	B-7
B. 6 Basis data prod string.....	B-8
B. 7 Basis data PPDM38 well daily data.....	B-9
B. 8 Basis data PPDM38 pden flow measurement	B-11
B. 9 Basis data PPDM38 well deferment daily	B-13
B. 10 Basis data PPDM38 pden vol prod day	B-14
B. 11 Basis data PPDM38 prod string alias	B-15
B. 12 Basis data PPDM38 pden vol prod defer day	B-16
B. 13 Basis data time dim.....	B-18
B. 14 Basis data string status hist	B-18
B. 15 Basis data PPDM38 prod str stat hist.....	B-19
B. 16 Basis data well job.....	B-20
B. 17 Basis data PPDM38 work order	B-21
B. 18 Basis data PPDM38 work order component	B-22
B. 19 Basis data PPDM38 vt prod string	B-25
B. 20 Basis data KLO asset doc.....	B-26
B. 21 Basis data KLO asset com.....	B-27
B. 22 <i>View</i> v dsc lpo summary wo shutin	B-28
B. 23 <i>View</i> v dsc lpo eventcat wo shutin.....	B-30
B. 24 <i>View</i> v dsc prodeff field wo shutin	B-31
B. 25 <i>View</i> v dsc lpo area wo shutin	B-32
B. 26 <i>View</i> KLO v dsc well exception	B-33
B. 27 <i>View</i> KLO v dsc today lpo summary area.....	B-35




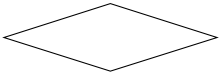
B. 28 <i>View</i> KLO v dsc ytd lpo summary area	B-37
B. 29 <i>View</i> KLO v dsc well test linreg eq	B-38
B. 30 <i>View</i> KLO v last well test.....	B-39
B. 31 <i>View</i> v dsc well status day.....	B-40

DAFTAR ISTILAH

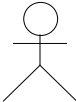


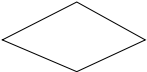


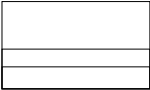
<i>BFPD</i>	= <i>Barrel fluid per day.</i>
<i>BOPD</i>	= <i>Barrel oil per day.</i>
<i>BOEPD</i>	= <i>Barrel oil each per day.</i>
<i>BWPD</i>	= <i>Barrel water per day.</i>
<i>Database</i>	= Basis data yang berisi kumpulan data-data hasil pengamatan.
<i>Data Dictionary</i>	= Kamus data untuk merancang tabel basis data.
<i>Form</i>	= Bentuk dari sebuah tampilan.
<i>Implementasi</i>	= Pelaksanaan atau penerapan.
<i>Input</i>	= Data yang dimasukkan.
<i>Interface</i>	= Tampilan antar muka.
<i>LPO</i>	= <i>Loss production day.</i>
<i>Output</i>	= Data yang dihasilkan.
<i>Proses</i>	= Runtunan perubahan dalam perkembangan sesuatu.
<i>MSCFD</i>	= <i>Million standard cubic feet per day</i> (gas).
<i>User</i>	= Pemakai atau pengguna sistem.
<i>YTD</i>	= <i>Year to date.</i>

DAFTAR SIMBOL

Keterangan notasi simbol *flowchart* :

	Mulai dan akhir program
	Proses
	Data
	Keputusan

Keterangan notasi simbol UML :

	Pengguna sistem (aktor)
	Proses (<i>use case</i>)
	Aktivitas (<i>activity</i>)
	Keputusan (<i>decision</i>)
	Status mulai (start state)
	Status selesai (final state)
	Kelas (<i>class</i>)

BAB I

PENDAHULUAN

1.1. Latar Belakang

Visualisasi merupakan upaya manusia dalam merepresentasikan maksud tertentu menjadi sebuah bentuk informasi yang lebih mudah dipahami. Pada zaman sekarang manusia melakukan pemrosesan visualisasi menggunakan komputer. Visualisasi berkembang dengan perkembangan teknologi, diantaranya rekayasa, visualisasi desain produk, pendidikan, multimedia interaktif dan kedokteran. Visualisasi mendukung pengambilan keputusan dan menganalisis data yang ditampilkan agar dapat memprediksi kesimpulan.

Dalam penelitian ini, penulis melakukan penelitian mengenai *Treemap* di PT. Chevron Pacific Indonesia (PT. CPI). Sebagai perusahaan energi terbesar PT.CPI memerlukan adanya pendataan mengenai setiap kegiatan eksplorasi dan produksi yang dihasilkan, seperti data minyak, gas, air, *steam*, *fluid*, *LPO (loss production oil)*, status *well*, data eksplorasi dan produksi lainnya yang ada pada setiap *well* di seluruh area dan daerah operasi PT. Chevron Pacific Indonesia. Data hasil eksplorasi dan produksi yang tersedia akan diproses dan dikalkulasikan sesuai kriteria datanya. Dengan banyaknya data yang berasal dari berbagai sumber data diperlukan proses untuk mengintegrasikannya dalam sebuah kesatuan data.

Agar hasil data yang telah terintegrasi dapat mudah dilihat, dipahami dan cepat dalam mengambil keputusan, maka dibutuhkan sebuah sistem yang dapat memvisualisasikannya dalam bentuk peta berdasarkan kebutuhan setiap *user*. Data-data ini akan diperbaharui setiap harinya untuk memantau status *well* yang ada dilapangan. Sistem yang akan dibangun ini diharapkan dapat menampilkan data berdasarkan *indicator* yang dipilih oleh *user* pada sistem. Sistem ini diharapkan juga dapat diakses oleh banyak *user* dimanapun berada selama terhubung dengan jaringan komputer perusahaan.

Berdasarkan penelitian Aimi Kobayashi, dkk, pada tahun 2012 yang berjudul *Edge Equalized Treemaps*, penelitian tersebut menggunakan *Treemap* dengan representasi yang ditandai dengan persegi panjang dengan ukuran *leaf*

yang sama, interval skala grafik dalam persegi panjang dapat bersatu, yang berarti seseorang dapat membandingkan grafik hanya dengan melihat gambarnya. Berdasarkan penelitian lain yaitu penelitian Abon Chaudhuri, dkk, pada tahun 2009 yang berjudul *A Self Adaptive TreeMap Based Technique for visualizing Hierarchical Data in 3D*, penelitian tersebut membangun visualisasi hirarki data *Treemap* berbasis 3D yang struktur informasinya dapat digunakan secara luas, visualiasi ini dapat menangani masalah seperti tampilan yang berantakan dan kurang halusny dalam men-*zooming*. Berdasarkan penelitian Chris Muelder, dkk, pada tahun 2008 yang berjudul *A Treemap Based Method for Rapid Layout of Large Graphs*, pada penelitian tersebut tata letak grafik dapat ditampilkan secara efektif dan cepat serta konsep tata letak metode *Treemap* sangat fleksibel. Dan berdasarkan penelitian Ben Shneiderman pada tahun 1991 yaitu yang berjudul *Tree Visualization with Tree-maps: A 2-d space-filling approach*, visualisasi 2D dalam mengisi ruang dengan metode *Treemap* berjalan dengan baik dan cepat, pewarnaan dan gambaran singkat mengenai ukuran dari *node* yang secara jelas memberikan kemudahan kepada pengguna.

Metode dasar yang sering digunakan dalam pelaporan analisis pemetaan data dalam bentuk tampilan visual x dan y adalah grafik batang. Grafik batang digunakan untuk perbandingan numerik dengan nilai-nilai partisi data, namun biasanya terbatas untuk beroperasi pada kumpulan data yang sangat kompleks, serta grafik batang hanya dapat menampilkan informasi sederhana (Mao Lin Huang, dkk, 2009). Visualisasi data berupa tabel, grafik garis dan grafik *pie* merupakan alat yang penting, namun untuk kegiatan yang kompleks dimana terdapat berbagai jenis kategori data perusahaan, jenis visualisasi ini tidak mendukung dan sangat sulit dalam menginterpretasikan data sesuai kebutuhan perusahaan (Ben Shneiderman, 2006). Tabel dan *spreadsheet* hanya dapat menampilkan 30-60 baris pada layar pada satu waktu, sedangkan presentasi warna-warna menggunakan *Treemap* dapat menampung ratusan atau ribuan data *item* dalam menampilkan data yang terorganisir. Metode *Treemap* sangat sesuai dengan penelitian yang dilakukan karena dapat berkembang dalam kebutuhan perusahaan yang memerlukan pemantauan data harian yang kompleks, dapat

memvisualkan data berdasarkan *group* data dan kategori warna dengan menyeluruh dan detail untuk setiap kondisi *well* yang terjadi.

Dengan menggunakan *Treemap*, para *user* (*manager* dan pimpinan tim) diharapkan dapat mengambil keputusan secara cepat agar tidak terjadinya hal-hal yang tidak diinginkan yang dapat mengganggu kinerja perusahaan dan merugikan perusahaan. Seperti terjadinya kehilangan produksi (*LPO*), matinya *well* produksi, status *well* dan melihat hasil dari produksi minyak dan gas.

1.2. Rumusan Masalah

Berdasarkan dari latar belakang yang telah dijelaskan, yang menjadi permasalahan adalah “Bagaimana mengintegrasikan data eksplorasi dan produksi dan memvisualisasikannya dengan menggunakan metode *Treemap* agar mudah dilihat, dipahami dan cepat dalam mengambil keputusan?”

1.3. Batasan Masalah

Dalam melakukan suatu penelitian diperlukan batasan-batasan agar tidak menyimpang dari yang telah direncanakan, sehingga tujuan yang sebenarnya dapat dicapai. Batasan-batasan dalam penelitian Tugas Akhir ini adalah:

1. Visualisasi *Treemap* yang digunakan yaitu berbasis 2D.
2. Kriteria warna sudah di tentukan dari awal perancangan.
3. Area lokasi operasi yang digunakan adalah KLO (*Kalimantan Operation*).
4. Data yang digunakan yaitu data eksplorasi dan data produksi.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai penulis dari Tugas Akhir ini adalah untuk dihasilkannya sebuah aplikasi yang dapat mengintegrasikan data dan memvisualisasikannya menggunakan metode *Treemap* di PT. Chevron Pacific Indonesia.

1.5. Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini adalah sebagai berikut:

BAB I. PENDAHULUAN

Membahas mengenai deskripsi umum dari Tugas Akhir yang meliputi latar belakang, rumusan masalah, hal-hal yang menjadi batasan masalah, tujuan penulisan, metodologi penulisan dan sistematika penulisan.

BAB II. LANDASAN TEORI

Membahas teori-teori pendukung. Teori yang diangkat yaitu mengenai integrasi data, visualisasi dan metode *Treemap*.

BAB III. METODOLOGI PENELITIAN

Membahas tahapan penelitian, pengumpulan data, analisa kebutuhan, perancangan perangkat lunak, implementasi, pengujian sistem dan kesimpulan akhir.

BAB IV. ANALISA DAN PERANCANGAN

Membahas tentang analisa sistem lama dan sistem baru dengan dibangun suatu rancangan integrasi data dan visualisasi menggunakan metode *Treemap*.

BAB V. IMPLEMENTASI DAN PENGUJIAN

Membahas mengenai implementasi integrasi data dan visualisasi menggunakan metode *Treemap* serta kesimpulan dari pengujian.

BAB VI. PENUTUP

Bab ini berisikan kesimpulan dari Tugas Akhir yang dibuat dan menjelaskan saran-saran Penulis kepada pembaca agar penerapan metode *Treemap* dapat dikembangkan lagi.

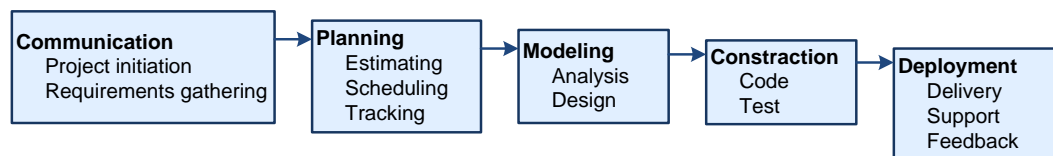
BAB II

LANDASAN TEORI

2.1 Model *Waterfall*

Menurut Roger S. Pressman (2010:39), “*Waterfall* Model dapat juga disebut sebagai *Classic Life Cycle*. Menunjukkan sebuah pendekatan sistematis untuk pengembangan perangkat lunak. Diawali dengan *communication*, *planning*, *modelling*, *construction* dan *deployment*”.

Tahapan dalam model *waterfall* menurut Pressman:



Gambar 2. 1 Tahapan dalam *waterfall*

1. *Communication*

Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer* atau perusahaan, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

2. *Planning*

Proses *planning* merupakan lanjutan dari proses *communication* (*analysis requirement*). Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang akan dilakukan.

3. *Modelling*

Proses *modelling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface* dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

4. *Construction*

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

5. *Deployment*

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

2.2 Integrasi Data

Menurut Han dan Kamber (2006), integrasi data adalah penggabungan data dari berbagai sumber penyimpanan data untuk menjadi satu kesatuan yang koheren. Jadi integrasi data merupakan suatu proses menggabungkan atau menyatukan data yang berasal dari sumber yang berbeda dan mendukung pengguna untuk melihat kesatuan data.

Menurut buku *The Gama Guide to The Data Management Body of Knowledge* (2009:16), “*Data integration: every data management function contributes to and benefits from data integration techniques, managing data assets through minimizing redudancy, consolidating data from multiple resource, and ensuring consistency across controlled redundant data with a “golden version”*”.

2.3 Sistem Basis Data

Sistem terdiri dari bagian-bagian yang saling terhubung untuk beroperasi bersama-sama untuk mencapai maksud tertentu, yang sering disebut dengan tujuan (*goal*) atau sasaran (*objective*). Jadi bagian-bagian di dalam sistem saling melengkapi karena satu maksud, tujuan dan sasaran.

Sistem basis data terdiri dari dua pengertian yaitu sistem dan basis data. Ada beberapa pendapat tentang definisi sistem, perbedaan pengertian ini tetap mengacu pada satu pengertian, antara lain:

- a. Sistem adalah sekelompok komponen yang terdiri dari manusia atau bukan manusia (*non human*) yang diorganisir dan diatur sedemikian rupa sehingga komponen-komponen tersebut dapat bertindak sebagai satu kesatuan dalam mencapai tujuan, sasaran bersama atau hasil akhir. Pengertian ini, mengandung arti pentingnya aspek pengaturan dan pengorganisasian komponen dari suatu sistem untuk mencapai sasaran bersama, karena bila tidak ada sinkronisasi dan koordinasi yang tepat, maka kegiatan masing-masing komponen, sub-sistem, atau bidang dalam suatu organisasi akan kurang saling mendukung (H.Kerzner).
- b. Sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, bekerja sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Kristiyanto, 2003).
- c. Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan (Kristiyanto, 2003).

Sedangkan Basis data dapat didefinisikan dalam sejumlah sudut pandang, seperti:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisir sedemikian rupa agar kelak dimanfaatkan kembali dengan cepat dan mudah.

- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu untuk memenuhi berbagai keperluan.
- c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data hanya sebuah objek yang pasif/mati. Basis data ada karena pembuatnya, karena itu basis data tidak akan pernah berguna jika tidak ada pengelola/penggerakannya dan yang menjadi penggerakannya secara langsung adalah program/aplikasi (*software*). Gabungan dari basis data dan penggerakannya menghasilkan sebuah sistem/aplikasi.

Sistem basis data merupakan kumpulan basis data yang secara umum mempunyai hubungan sistem.

2.3.1. Penyusunan Sistem Basis Data

Sistem basis data merupakan lingkup yang lebih luas dari basis data. Sistem basis data merupakan sekumpulan basis data yang secara umum mempunyai hubungan. Sistem basis data dapat mencakup semua bentuk komponen data yang ada dalam suatu sistem. Sedangkan sistem basis data merupakan komponen utama yang menyusun sistem basis data. Adapun komponen-komponen pembentuk sistem basis data adalah sebagai berikut:

- a. *Bit* adalah suatu sistem angka biner yang terdiri atas dua macam nilai, yaitu 0 dan 1.
- b. *Byte* adalah bagian terkecil yang dapat dialamatkan dalam memori.
- c. Rinci data atau *item* adalah unit terkecil yang disebut data.
- d. *Agregat* data adalah kumpulan rinci data yang mempunyai ciri tertentu dan diberi nama.
- e. *Record* (rekaman) merupakan sekumpulan data *item* atau *agregat* data yang saling berhubungan dengan suatu objek tertentu.
- f. *File* adalah kumpulan *record* sejenis secara relasi.
- g. Basis data adalah sekumpulan dari bermacam-macam tipe *record* yang mempunyai hubungan antar *record*, *agregat* data dan rinci data terhadap objek tertentu.
- h. Sistem basis data

2.3.2. Tujuan Basis Data

Pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan. Tujuan tersebut sebagai berikut:

a. Kecepatan dan Kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, dari pada menyimpan data secara manual.

b. Efisiensi Ruang Penyimpanan (*Space*)

Karena keterkaitan yang antar kelompok data dalam sebuah basis data, maka *redundancy* data pasti akan selalu ada. Dengan basis data efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan karena basis data dapat melakukan penekanan jumlah *redundancy* data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi antar kelompok data yang saling berhubungan.

c. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tiap data, domain data, keunikan data yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekankan ketidakakuratan/penyimpanan data.

d. Ketersediaan (*Availability*)

Data yang sudah jarang atau bahkan tidak pernah lagi digunakan, dapat diatur untuk dilepaskan dari sistem basis data yang aktif baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan. Dengan pemanfaatan teknologi jaringan komputer, data yang berada di suatu lokasi/cabang dapat diakses (menjadi tersedia/*available*) bagi lokasi/cabang lain.

e. Kelengkapan (*Completeness*)

Lengkap tidaknya data yang dikelola dalam sebuah basis data bersifat relatif (baik terhadap kebutuhan pemakai maupun terhadap waktu). Dalam sebuah basis data, disamping data juga harus menyimpan struktur yang

mendefinisikan objek-objek dalam basis data maupun *define* detail dari tiap objek seperti struktur *file*/tabel atau indeks.

f. Keamanan (*Security*)

Keamanan data diperlukan untuk melindungi data terhadap akses yang tidak legal oleh pihak-pihak yang tidak berwenang yang bermaksud merugikan atau bahkan merusak data dalam basis data atau kerusakan. Keamanan merupakan aspek krisis lain dalam sistem basis data. Keamanan data juga diperlukan untuk melindungi data dari kerusakan-kerusakan.

g. Kebersamaan (*Sharebility*)

Yang dimaksud adalah bahwa sistem basis data yang dikembangkan harus dapat digunakan oleh pemakai-pemakai yang berbeda atau grup-grup pemakai yang berbeda dapat menggunakan data yang sama dalam basis data. Hal ini penting karena dalam basis data akan digunakan oleh beberapa pihak yang berbeda yang berkepentingan terhadap data tersebut.

Sistem basis data merupakan lingkup yang lebih luas dari basis data. Sistem basis data merupakan kumpulan basis data yang secara umum mempunyai hubungan sistem. Sistem basis data dapat mencakup semua bentuk komponen data yang ada dalam suatu sistem. Sedangkan basis data merupakan komponen utama menyusun sistem basis data.

Database dan teknologi *database* memiliki pengaruh yang besar terhadap perkembangan penggunaan komputer. Data adalah fakta yang dapat direkam dan memiliki arti secara implisit. Sebagai contoh nama, nomor telepon dan alamat. *Database* adalah sekumpulan data yang saling berhubungan dan memiliki arti secara implisit.

Database Management System (DBMS) adalah sekumpulan program yang memungkinkan pengguna untuk membuat dan memelihara (*maintenance*) suatu *database*. Dengan kata lain, DBMS merupakan *general-purpose software system* dan *sharing database* antara berbagai pengguna dan aplikasi. Beberapa fungsi penting yang disediakan oleh DBMS adalah keamanan dan pemeliharaan (*maintenance*). *Database system* tidak hanya berupa *database* itu sendiri tetapi juga terdiri dari definisi dan deskripsi struktur dan batasan-batasan (*constraint*)

dari *database*. Definisi disimpan dalam katalog DBMS, yang berisi informasi berbagai batasan terhadap data tersebut. Informasi yang disimpan pada katalog DBMS disebut juga *meta-data* yang menjelaskan struktur utama *database*.

2.3.3. Komponen-komponen Sistem Basis Data

Komponen-komponen sistem basis data antara lain sebagai berikut:

- a. Perangkat Keras (*Hardware*)
Perangkat keras dalam suatu sistem basis data mempunyai komponen-komponen utama yang berupa unit pusat pengolah CPU (*Central Processing Unit*) dan unit penyimpanan (*Storage Unit*).
- b. Sistem Operasi (*Operating System*)
Sistem operasi merupakan program yang mengaktifkan/memfungsikan sistem komputer, mengendalikan sumber daya (*resource*) dalam komputer dan melakukan operasi-operasi dasar komputer (operasi *I/O*, pengelolaan *file* dan lain-lain).
- c. Basis data (*Database*)
Basis data dapat berisi/memiliki objek basis data (seperti *file*, tabel indeks dan lain-lain).
- d. Aplikasi atau Perangkat Lunak Pengolah Basis Data (*DBMS*)
Suatu pengolahan basis data yang secara fisik tidak langsung dilakukan oleh pemakai tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus/spesifik. *DBMS* adalah perangkat lunak yang memberikan fasilitas (yang tersedia dan dapat digunakan) untuk melakukan fungsi pengaturan, pengawasan, pengendalian/*control*, pengolahan dan koordinasi terhadap semua proses atau operasi yang terjadi pada sistem basis data.
- e. Pengguna (*User*)
Orang yang menggunakan atau mengoperasikan komputer (menjalankan aplikasi/perangkat atau berinteraksi terhadap sistem).

2.4 Oracle

Oracle merupakan RDMS yang paling banyak digunakan oleh perusahaan-perusahaan di dunia. Sejarah perkembangannya yang cukup panjang telah membawa Oracle menjadi *database* yang sangat handal sehingga menjadikannya sebagai *database* yang paling banyak digunakan.

2.4.1. Oracle dan ORDBMS

Oracle merupakan *database* yang menggunakan konsep RDBMS. Pada RDMBS, sebuah tabel merupakan struktur penyimpanan dasar. Satu tabel atau lebih membentuk sebuah *relational database*.

Oracle 10g merupakan *database* yang pertama kali didesain untuk aktivasi *Enterprise Grind Computing*. Oracle 10g mampu memotong biaya operasional selain memberikan *service* berkualitas tinggi, juga mampu secara cepat merespon terhadap kebutuhan bisnis dengan resiko yang rendah. Selain itu Oracle 10g juga mudah untuk di *deploy* dan di *maintenance*. Oracle 10g memberikan kinerja industri terbaik, skalabilitas, keamanan dan kehandalan yang didukung Windows, Linux, Unix dan Solaris. Ini memberikan fitur yang lengkap agar dengan mudah mengatur proses pengolahan transaksi yang menuntut intelijen bisnis dan aplikasi manajemen konten.

Oracle merupakan *database* yang berkemampuan menangani objek yang lebih dikenal dengan ORDBMS (*Object Relational Database Management System*). Beberapa fitur yang dimiliki *database* Oracle yaitu sebagai berikut:

1. Menyimpan *user defined data type* sebagai objek tabel.
2. *Method* yang dapat diasosiasikan ke dalam definisi tabel.
3. Kolom dapat menggunakan tipe data yang didefinisikan oleh *user*.

2.4.2. Kehandalan Database Oracle

Beberapa keunggulan *database* Oracle adalah sebagai berikut:

1. *Scability*, kemampuan menangani banyak *user* yang melakukan koneksi secara simultan tanpa berkurangnya *performance* secara signifikan. Dalam dokumentasi Oracle disebutkan bahwa *database* Oracle sanggup melayani puluhan ribu *user* secara simultan.

2. *Realibility* yang bagus, yakni kemampuan untuk melindungi data dari kerusakan jika terjadi kegagalan fungsi pada sistem seperti *disk failure*.
3. *Serviceability*, yaitu kemampuan untuk mendeteksi masalah, kecepatan dalam mengoreksi kesalahan dan kemampuan melakukan konfigurasi ulang struktur data.
4. *Stability*, yaitu kemampuan untuk tidak *crash* karena beban kerja yang tinggi. Hal ini berkaitan dengan *scalability*.
5. *Avaibility*, yaitu kemampuan dalam penanganan *crash* atau *failure* agar *service* dapat tetap berjalan. Misalnya saja dengan tersedianya fasilitas pendistribusian *database* pada beberapa data server dan juga pemulihan *database* (*recovery*).
6. *Multiplatform*, yaitu dapat digunakan pada banyak sistem operasi seperti Windows, Unix, Linux dan Solaris.
7. Mendukung data yang sangat besar. Menurut dokumentasi Oracle, dapat menampung sampai 512 petabytes (1 petabytes = 1.000.000 gigabytes).
8. Sistem *security* yang cukup handal
9. Mendukung *database* berorientasi objek.
10. Dapat menampung hampir semua tipe data seperti teks, *image*, *sound*, *video* dan *time series*.

Keunggulan-keunggulan di atas membuat Oracle banyak digunakan pada aplikasi internet aplikasi *client server* pada *local area network* (LAN).

2.4.3. SQL

SQL merupakan bahasa yang digunakan oleh banyak aplikasi atau *tool* untuk berinteraksi dengan *server database*. SQL adalah bahasa fungsional yang tidak mengenal iterasi dan tidak bersifat *procedural*. SQL menggunakan perintah-perintah dengan kata-kata sederhana dan mirip dengan bahasa manusia sehari-hari, tentu saja dalam bahasa inggris. Pada SQL, *user* hanya menentukan apa hasil akhir yang diinginkan.

2.4.4. PL/SQL

PL/SQL adalah bahasa *procedural* yang dikembangkan sendiri oleh *Oracle Corporation* untuk mengoptimalkan pembuatan aplikasi *database* yang menggunakan *database* Oracle. Kata PL pada PL/SQL merupakan singkatan dari *Procedural Language*. Pada PL/SQL ditambahkan beberapa hal yang dikenal pada dunia pemrograman, seperti *variable*, *loop*, pemrosesan berdasarkan kondisi, operasi *cursor*, modularasi dan lain-lain. Semua tambahan itu bertujuan untuk meningkatkan kinerja operasi-operasi SQL pada *database* Oracle sehingga manfaat dan keandalannya menjadi lebih maksimal.

Berikut ini merupakan keuntungan penggunaan PL/SQL, yaitu sebagai berikut:

1. Mudah dipelajari dan digunakan.
2. Dapat disimpan di *database* Oracle.
3. Terintegrasi dengan pada *database* Oracle.
4. Kemampuan yang bagus untuk menangani blok data yang besar.
5. Terdapat banyak *package* yang disediakan Oracle untuk meningkatkan *performance*-nya.
6. Mendukung program unit dengan ataupun tanpa nama.
7. Mendukung enkapsulasi dan modularasi.
8. Mendukung *overloading*.
9. Mendukung *advanced* dan *type*.
10. Mengenal pengelompokan *procedure* dan *function* ke dalam satu kesatuan *package*.

2.5 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem peranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

2.5.1. Tujuan *Unified Modelling Language* (UML)

UML dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan manapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *modelling* aplikasi *procedural* dalam VB atau C.

2.5.2. Konsep Dasar UML

UML mendefinisikan diagram-diagram sebagai berikut:

1. *Usecase Diagram*

Usecase diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Usecase diagram* menekankan pada apa yang diperbuat sistem, dan bukan bagaimana sistem membuat sesuatu. Sebuah *use case* mempresentasikan sebuah interaksi antar aktor dengan sistem.

Use case merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja dan sebagainya. Seorang aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Usecase diagram* sangat membantu dalam menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan *client* dan merancang *test case* untuk semua *feature* yang ada pada sistem.

2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain.

3. *Statechart Diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* yang lainnya) suatu objek pada sistem sebagai akibat aksi yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

4. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivasi dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di *trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

5. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

6. *Collaboration Diagram*

Collaboration Diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing

objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1. *Message* dari level 1 yang sama memiliki prefiks yang sama.

7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan serta ketergantungan (*dependency*) antar komponen piranti lunak. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

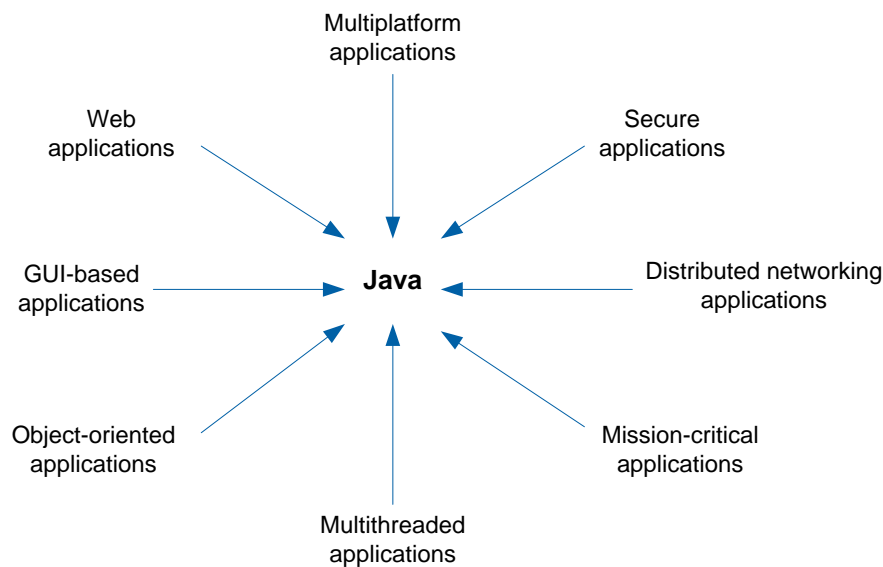
8. *Deployment Diagram*

Deployment/physical diagram menggambarkan *detail* bagaimana komponen di *deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server* dan hal-hal lain yang bersifat fisikal.

2.6 Java

Java merupakan bahasa pemrograman berorientasi objek dan bebas *platform*, yang dikembangkan oleh insinyur *SUN Micro System* bernama James Gosling yang pada awalnya diberi Oak dan diubah menjadi Java dengan sejumlah keunggulan yang memungkinkan Java dijadikan sebagai bahasa pengembangan *enterprise* (Utdirartatmo, 2003).

Gambar di bawah ini memperlihatkan keunggulan-keunggulan Java yaitu sebagai berikut:



Gambar 2. 2 Keunggulan-keunggulan Java

Berikut ini merupakan penjelasan keunggulan-keunggulan yang dimiliki Java yang tampak pada gambar di atas yaitu:

1. Sederhana

Bahasa pemrograman Java cukup sederhana sebagai suatu bahasa pemrograman yang berorientasi objek. Sintaks untuk Java sama seperti sintaks bahasa pemrograman C++ yang sudah terlebih dahulu terkenal.

2. Berorientasi Objek

Konsep pemrograman berorientasi objek dirancang agar pemrograman dapat dilihat sebagai suatu kehidupan nyata. Ini membuat pengembang *software* menjadi lebih mudah karena seperti berhubungan dengan kehidupan nyata, bukan lagi kehidupan “dunia lain”.

3. *Multiplatform*

Java merupakan bahasa pemrograman yang dapat berjalan diberbagai macam sistem operasi seperti *Windows*, *Linux*, *Unix* dan *Macintosh* asalkan JVM (*Java Virtual Machine*) telah terinstal untuk *platform* yang bersangkutan. Java juga dapat digunakan untuk membuat program yang berjalan diatas berbagai *mobile device* (dengan J2ME), PDA, *embedded system* dan lain-lain.

4. Aplikasi Web

Java merupakan bahasa pemrograman yang memiliki dukungan sangat baik terhadap aplikasi web, karena pada awalnya Java dilahirkan sebagai solusi untuk menjawab kebutuhan bahasa pemrograman yang mendukung aplikasi berbasis jaringan. Diantara teknologi Java yang mendukung aplikasi web adalah Applet, JSP (untuk aplikasi web berbasis *server*), CORBA (untuk aplikasi terdistribusi) dan lain-lain.

5. Keamanan

Java sangat tepat digunakan untuk menangani kebutuhan aplikasi *enterprise* adalah keampuannya dalam soal keamanan. Fitur keamanan Java ada 2 paket, yaitu pada JDK (*Java Development*) dan pada *Java Cryptography Extension* (jCE). Fitur keamanan yang tersedia pada kedua paket tersebut meliputi *signature* (untuk mendatangi dokumen), *message digest*, meliputi pembangkitan kunci, autentikasi, enkripsi dan bilangan besar (*big number*).

6. Aplikasi Jaringan Terdistribusi

Java dapat digunakan pada aplikasi jaringan terdistribusi. Aplikasi yang berjalan pada jaringan terdistribusi melibatkan sejumlah komputer yang berkomunikasi secara transparan, yaitu seolah sejumlah komputer itu merupakan sebuah komputer yang bersatu untuk menjalankan sebuah perintah.

7. *Software Mission-Critical*

Java bisa digunakan untuk pengembangan *software* dimana tingkat *error* yang terjadi sangat diperhatikan, dimana sebuah *error* pada eksekusi program bisa mengakibatkan kerusakan fatal pada sistem, secara keseluruhan. Sebagai dukungan terhadap *software* seperti ini, Java menghilangkan sejumlah fitur C++ yang berpotensi menghasilkan *error* yang fatal, seperti *pointer*, konversi tipe tanpa dicek dan lain-lain.

8. *Multithread*

Fitur *multithread* digunakan untuk menjalankan sejumlah proses secara bersamaan.

2.7 Treemap

Treemap merupakan metode untuk menampilkan struktur data susunan yang sangat besar dengan menggunakan bentuk persegi panjang (*rectangles*) dalam *space* (ruang) kecil (Mark Bruls, 2010). *Space* (ruang) dalam memvisualisasinya dibagi menjadi empat persegi panjang (*rectangles*) dimana ukuran dan *ordered*-nya berdasarkan variable kuantitatif. *Treemap* menampilkan persegi panjang 2D dengan mengisi *space* hingga 100 % dari tampilan *space* yang digunakan (Ben Shneiderman, 2007). Dalam *node*, atribut merupakan hal penting, karena memberikan tampilan area yang lebih. Pengguna dapat memilih untuk menentukan atribut *node leaf* yang menentukan alokasi *space*.

Tingkatan dalam *Treemap* yaitu memvisualisasikan persegi panjang utama yang mengandung persegi panjang lainnya. Setiap penentuan persegi panjang pada tingkat yang sama dalam susunan mewakili sebuah kolom atau ekspresi dalam data tabel. Setiap persegi panjang pada tingkat dalam susunan mewakili kategori dalam sebuah kolom. Sebagai contoh, sebuah persegi panjang utama mewakili benua. Setiap persegi panjang yang lainnya mewakili negara yang memuat persegi panjang yang lebih kecil yang mewakili kota-kota di negara-negara tersebut (Mark Bruls, 2010).

2.7.1. Sejarah Treemap

Area berbasis visualisasi sudah ada selama beberapa dekade yang lalu. *Mosaic Plot* dan diagram *Marimekko* keduanya menggunakan persegi panjang *Tilings* untuk menampilkan kontribusi yang digabungkan. Fitur pembeda *Treemap* adalah kontruksi rekursif yang memungkinkan untuk diperluas ke susunan data dengan sejumlah tingkat. Ide ini ditemukan di *HCIL (Human Computer Interaction Lab) University of Maryland, Collage Park* oleh *Professor Ben Shneiderman* di awal tahun 1990-an (Ben Shneiderman, dkk, 2010). Shneiderman dan rekan-rekannya kemudian memperdalam ide tersebut dengan memperkenalkan variasi teknik interaktif untuk memfilter dan menyesuaikan *Treemap*.

Treemap pada awalnya menggunakan algoritma *Tilling* yang sederhana dengan menggabungkan kumpulan data. *Treemap* memiliki sifat yang stabil dan

dapat diurutkan. Metode penggabungan kumpulan data sering menghasilkan pengolahan data yang panjang dan gambar persegi panjang yang kecil-kecil. Pada tahun 1994 *Hascoet* dan *Beaudouin Lafon* menciptakan algoritma “*Squerifying*”, kemudian dipopulerkan oleh *Jerke Van Wijk*, yang menciptakan *Tillings* dengan persegi panjang mendekati *square*. Pada tahun 1999, *Martin Wattenberg* menggunakan variasi dari algoritma “*Squerifying*” yang kemudian disebut “*Pivot* dan *Slice*”, untuk menciptakan *Treemap* pertama yang berbasis Web, yaitu *The Smart Money Map of the Market*, yang datanya ditampilkan pada ratusan perusahaan di pasar saham Amerika. Setelah peluncurannya, *Treemap* mendapat banyak peminat, terutama dalam hal keuangan.

Gelombang ketiga dari inovasi *Treemap* datang sekitar tahun 2004, setelah *Marcos Weskamp* membuat *Newsmap*, yaitu sebuah *Treemap* yang menampilkan *news headline*. Dalam beberapa tahun terakhir, *Treemap* telah membuat jalan mereka ke arah *mainstream media*, termasuk yang digunakan oleh *New York Times*.

2.7.2. Metode *Treemap*

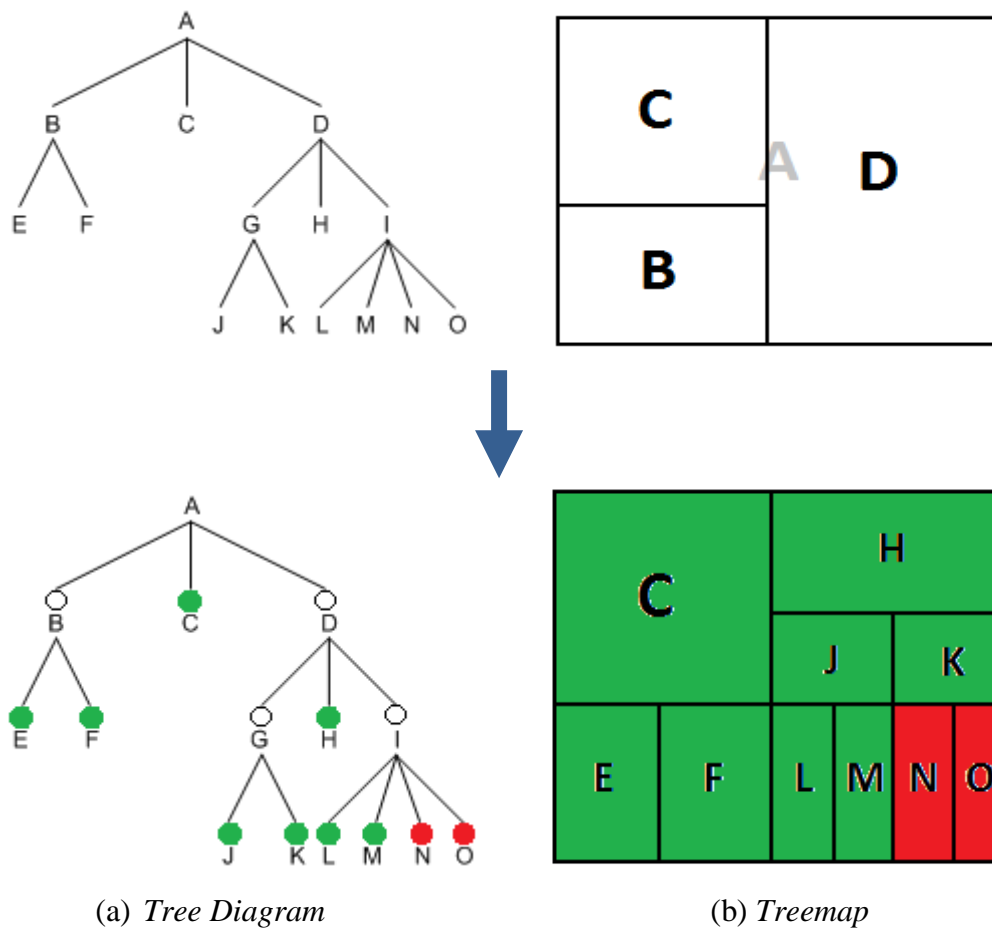
Berdasarkan penelitian Aimi Kobayashi, Kazuo Misue dan Jiro Tanaka, pada tahun 2012 yang berjudul *Edge Equalized Treemaps*, penelitian tersebut menyatukan lebar sumbu horizontal masing-masing grafik yang ditandai dengan persegi panjang dengan ukuran *leaf* yang sama. Ketinggian masing-masing *leaf* persegi panjang dalam *Treemap* disamakan sesuai dengan kebutuhan data. Interval skala grafik dalam persegi panjang dapat bersatu, yang berarti seseorang dapat membandingkan grafik hanya dengan melihat gambarnya.

Berdasarkan penelitian Abon Chaudhuri dan Hai-Wen Shen pada tahun 2009 yang berjudul *A Self Adaptive TreeMap Based Technique for visualizing Hierarchical Data in 3D*, penelitian tersebut membangun visualisasi hirarki data *Treemap* berbasis 3D yang struktur informasinya dapat digunakan secara luas, visualiasi ini dapat menangani masalah seperti tampilan yang berantakan dan kurang halusny dalam men-*zooming*. *Treemap* memanfaatkan teknik fleksibilitas yang bertujuan untuk mencegah adanya *occlusion*, yaitu apabila peta terangkak ada beberapa bagian peta yang berada didasar ikut terangkak, ini merupakan masalah yang sering terjadi pada visualisasi 3D.

Berdasarkan penelitian Chris Muelder dan Kwan-Liu Ma pada tahun 2008 yang berjudul *A Treemap Based Method for Rapid Layout of Large Graphs*, pada penelitian tersebut tata letak grafik dapat ditampilkan secara efektif dan cepat serta konsep tata letak metode *Treemap* sangat fleksibel. Metode ini dapat digunakan sebagai masukan algoritma yang berulang, yaitu dimana dapat mengurangi jumlah iterasi yang dibutuhkan untuk meng-*converge* tata letak optimal yang paling dekat.

Dan berdasarkan penelitian Ben Shneiderman pada tahun 1991 yaitu yang berjudul *Tree Visualization with Tree-maps: A 2-d space-filling approach*, Ben ingin mendapatkan representasi yang lebih baik dari pemanfaatan penyimpanan pada *hardisk* yang dilihat dari perspektif direktori level *multiple* subdirektori dan *files*, seperti di Unix, Macintosh Finder atau Ms-Dos. Visualisasi 2D dalam mengisi ruang dengan metode *Treemap* untuk memvisualisasikan berjalan dengan baik dan cepat. Pewarnaan dan gambaran singkat mengenai ukuran dari *node* yang secara jelas memberikan kemudahan kepada pengguna.

Ada banyak metode untuk menelusuri dan menampilkan struktur susunan informasi, atau untuk jangka pendek, seperti *tree* (Mark Bruls, dkk, 2010). *File browser* merupakan contoh yang paling terkenal. Biasanya *list* dari *file* dan direktori yang digunakan dalam susunan ditampilkan berdasarkan ukuran dan kategori datanya. Jumlah *file* dan direktori dapat ditampilkan secara bersamaan (memiliki *limit* dalam menampilkannya), yang mana seseorang dapat tahu apa yang harus dicarinya.



Gambar 2. 3 Tree diagram dan Treemap

Ada banyak cara untuk memvisualisasikan struktur tersebut secara efektif. Seperti gambar diatas (a), elemen-elemennya ditampilkan dengan *nodes*, relasinya ditunjukkan dari *link parent* ke *nodes child*-nya. Semakin majunya teknologi, ada banyak teknologi yang disajikan untuk meningkatkan efisiensi dan estetika teknologi, seperti diagram baik dalam 2D dan 3D. Diagram tersebut sangat efektif untuk *tree* yang kecil, namun biasanya akan mengalami kegagalan ketika lebih dari beberapa ratus elemen yang harus divisualisasikan secara bersamaan. Alasan utama dari *limit*-nya yaitu bahwa *node* dan *link* diagram menggunakan ruang *display* yang tidak efisien, sebagian besar menggunakan *pixel* sebagai *background*-nya.

Treemap dikembangkan untuk mengatasi masalah ini. Ditampilkan secara penuh yang digunakan untuk memvisualisasikan isi dari *tree*. Pada gambar diatas (b), setiap *node* (seperti yang ditunjukkan dalam diagram *tree*) memiliki nama (a)

dan ukuran yang terkait (angka). Ukuran *leaf* dapat mewakili ukuran *file* secara individual, ukuran *node* yang *non-leaf* adalah jumlah ukuran anak-anaknya. *Treemap* dibangun melalui pembagian rekursif dari persegi panjang yang awal. Ukuran masing-masing sub persegi panjang sesuai dengan ukuran *node*-nya. Tujuan dari *subdivision* pertingkatnya yaitu horizontal, vertikal, *next* vertikal dan lain sebagainya. Hasilnya persegi panjang yang awal dibagi menjadi persegi panjang yang lebih kecil, sehingga ukuran masing-masing persegi panjang mencerminkan dari ukuran *leaf*-nya. Struktur *tree* juga menggambarkan *Treemap* tersebut, sebagai hasil dari konstruksinya. Warna dan penjelasan dapat digunakan untuk memberikan informasi tambahan mengenai *leaf*.

Basic dari *Treemap* yaitu setiap persegi panjang adalah *data record*, ukuran persegi panjang tergantung pada nilai yang dipilih berdasarkan *size indicator*, ukuran persegi panjang pada *Treemap* ditampilkan mulai dari sudut kiri atas sampai ke sudut kanan bawah dengan persegi panjang terbesar diposisikan atau berada di sudut kiri atas dan yang terkecil berada di sudut kanan bawah dan warna persegi panjang tergantung dari indikator warna yang dipilih. *Treemap* dapat mengelompokkan *data record* yang sesuai berdasarkan kategorinya (Shinichiro Terashima, 2012)

Untuk susunannya, yaitu ketika persegi panjangnya bersarang (*nested*), penataannya sama dengan persegi panjang yang diulang untuk masing-masing persegi panjang lainnya di dalam *Treemap*. Hal ini berarti, bahwa ukuran dan posisinya mewakili persegi panjang dari sub persegi panjang lainnya yang ditentukan oleh jumlah area yang terkandung didalamnya.

2.8 VBScript

VBScript (*Visual Basic Scripting*) adalah sebuah bahasa *script* yang diinterpretasikan saat dieksekusi yang dikembangkan oleh *Microsoft Corporation* pada tahun 1996 yang sering digunakan oleh web *Internet Explorer*. VBScript merupakan sebuah bahasa *script* turunan dari bahasa pemrograman *Visual Basic for Application* (VBA) yang digunakan di dalam Microsoft Office dan beberapa *platform* pengembangan buatan Microsoft lainnya.

VBScript tidak dapat digunakan untuk membuat program yang berdiri sendiri. Akan tetapi, dimasukkan ke dalam sebuah berkas HTML. Ketika *Internet Explorer* membuka dokumen berkas HTML tersebut, VBScript dapat melakukan fungsi yang sama seperti Javascript, yang kemudian *script* tersebut akan dieksekusi.

VBScript memungkinkan pemakai melakukan interaksi dengan halaman web, bukan hanya menikmati informasi yang ditayangkannya. Ada beberapa skenario interaksi halaman web dengan pemainnya. Sebagai contoh, halaman web dapat menanyakan sesuatu kepada pemakai dan memberikan reaksi memberikan respon yang diberikan oleh pemakai. VBScript dapat mengambil masukan data dari pemakai, memeriksanya dan memastikan data tersebut valid atau memenuhi kriteria tertentu. Kemudian VBScript dapat menjalankan *server* untuk menyimpan data atau bereaksi sesuai dengan data yang dimasukkan pemakai.

2.9 Visualisasi

Menurut McCormick, et al, 1997, Visualisasi adalah sebagai berikut:

- i. Metode penggunaan komputer untuk mentransformasi simbol menjadi *geometric*.
- ii. Memungkinkan peneliti mengamati simulasi dan komputasi.
- iii. Memberikan cara untuk melihat yang tidak terlihat.
- iv. Memperkaya proses penemuan ilmiah dan mengembangkan pemahaman yang lebih dalam dan tak diduga.
- v. Dalam berbagai bidang telah merevolusikan cara ilmuwan meneliti sains.

Visualisasi adalah penggunaan komputer pendukung, penggambaran data visual interaktif untuk memperkuat pengamatan (Card, Mckinlay Shneiderman, 1998).

Jadi visualisasi merupakan rekayasa dalam pembuatan gambar, diagram atau animasi untuk penampilan suatu informasi, dan visualisasi data dapat mengkonversi data ke dalam format visual atau tabel sehingga karakteristik dari data dan relasi diantara *item* data atau atribut dapat dianalisa atau dilaporkan. Secara umum, visualisasi dalam bentuk gambar baik yang bersifat abstrak maupun

nyata telah dikenal sejak awal peradaban manusia. Contohnya seperti meliputi lukisan di dinding-dinding gua dari manusia purba, bentuk hirolif Mesir, sistem geometri yunani dan teknik pelukisan dari Leonardo Da Vinci untuk tujuan rekayasa dan ilmiah.

Manusia memiliki kemampuan membangun yang baik untuk menganalisa sejumlah besar informasi yang dipresentasi secara visual. Ia dapat mendeteksi pola umum dan *trend*, *outliers* dan pola yang tidak umum. Dalam visualisasi terdapat istilah representasi yang berarti pemetaan informasi menjadi format visual. Objek data, atribut-atributnya dan relasi diantara objek-objek data diterjemahkan ke dalam elemen grafis seperti titik, garis, bentuk-bentuk tertentu dan warna. Misalnya objek-objek sering dipresentasikan sebagai titik, nilai atribut-atributnya dapat direpresentasikan sebagai posisi dari titik-titik atau karakteristik dari titik sebagai contoh warna, ukuran dan bentuk. Jika posisi digunakan, maka relasi pada titik, apakah terbentuk dalam *group* atau sebuah titik *outliers*, dapat dengan mudah dilihat.

2.10 Pemrograman Web

Web adalah suatu metode untuk menampilkan informasi di internet, baik berupa teks, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui sebuah *browser*. Sebuah *web page* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu *protocol* yang menyampaikan informasi yang sangat besar. *Web page* layaknya sebuah buku yang dapat menampung berbagai informasi tentang banyak hal baik bersifat komersil maupun non komersil. Melalui media web ini seseorang dapat memberikan informasi tertentu kepada orang lain yang berada di seluruh dunia.

Bahasa pemrograman berbasis web antara lain Perl, Java Servelt, Asp, PHP dan sebagainya. Komponen pendukung lainnya adalah *markup language* (HTML, XML, WML) dan beberapa komponen skrip (*Javascript*, J-Query, VBScript, CGI Script) (Sunarfrihantono, 2002).

2.10.1. HTML

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman web. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language* (SGML). HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

HTML disebut *markup language* karena HTML berfungsi untuk memperindah *file* tulisan (teks) dapat dilihat pada *web browser* yang ada. Format HTML secara garis besar adalah sebagai berikut:

`<HTML>`

`<Head>`

..... *disini untuk menuliskan keterangan tentang dokumen*

`</Head>`

`<Body>`

..... *disini untuk menuliskan isi homepage*

`</Body>`

`</HTML>`

Head sebagai kepala dari suatu dokumen HTML yang berisi judul, alamat dasar dari dokumen HTML dan hubungan antar dokumen yang satu dengan yang lain. *Head* hanya berfungsi sebagai pemberi informasi dan pengendali sehingga informasi apapun yang ada dalam *head* tidak akan ditampilkan pada dokumen utama saat HTML dibuka suatu *browser*.

Body merupakan bagian utama suatu dokumen yang berisi semua teks yang ditampilkan melalui *browser*. Di dalam *body* terdapat teks dan kode-kode untuk memformat teks.

Selain *head* dan *body* ada bagian lain yang disebut *tag*. Setiap *tag* memiliki pembuka dan penutup. Berikut ini merupakan sintaks dalam penulisan *tag*.

`<tag>objek</tag>`

Atau

`<tag>`

Objek

`</tag>`

Tag merupakan kode-kode HTML yang digunakan untuk memberi format tertentu terhadap teks atau grafik sehingga akan tampak seperti yang diinginkan pada *layer*.

2.10.2. PHP

PHP (*hypertext preprocessor*) adalah bahasa pemrograman yang berfungsi untuk membuat *website* dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file* dan *folder*, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website*. Blog, Toko *Online*, CMS, Forum, dan *Website Social Networking* adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa *scripting*, bukan bahasa *tag-based* seperti HTML. PHP termasuk bahasa yang *cross-platform*, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file *plain text* (teks biasa) dan mempunyai akhiran “.php”.

Aturan penulisan script PHP adalah:

1. Semua script PHP harus diapit oleh tanda:
`<?php` dan `?>` , atau
`<script language='php'>` dan `</script>` , atau
`<? dan ?>` , atau
`<% dan %>`
2. Tetapi tanda yang resmi dan paling banyak digunakan adalah yang pertama, yaitu `<?php dan ?>`
3. Pada setiap akhir perintah, diakhiri dengan tanda titik koma (;)

Contoh:

```
<html>
  <head>
    <title>Contoh Sederhana </title>
  </head>
  <body>
    <?php echo("Hallo apa kabar? Nama saya PHP
      script"); ?>
  </body>
</html>
```

Dalam PHP setiap nama variabel diawali tanda *dollar* (\$). Misalnya nama variabel *a* dalam PHP ditulis dengan *\$a*. Jenis suatu variabel ditentukan pada saat jalannya program dan tergantung pada konteks yang digunakan.

Contoh:

```
<?php
  $a="5";
  $b="2";
  $hasil=$a+$b;
  echo($hasil);
?>
```

Pada PHP, tipe data variabel tidak didefinisikan oleh *programmer*, akan tetapi secara otomatis ditentukan oleh interpreter PHP. Namun demikian, PHP mendukung 8 (delapan) buah tipe data primitif, yaitu :

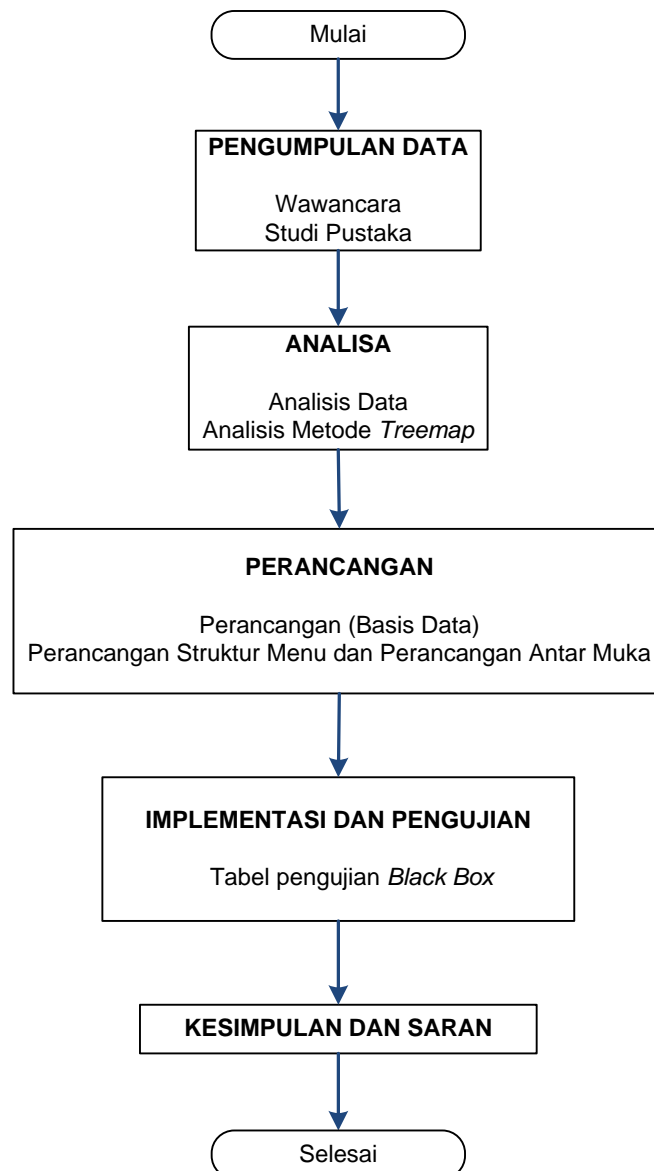
1. *Boolean*
2. *Integer*
3. *Float*
4. *String*
5. *Array*
6. *Object*
7. *Resource*
8. *Null*

BAB III

METODOLOGI PENELITIAN

3.1. Alur Metodologi Penelitian

Tahapan penelitian yang akan dilaksanakan pada penelitian ini dapat dilihat pada Gambar 3.1 sebagai berikut:



Gambar 3. 1 Alur metodologi penelitian

Gambar 3.1 di atas merupakan metodologi yang akan dilakukan oleh Penulis. Metodologi penelitian bertujuan untuk menguraikan kegiatan yang akan dilaksanakan selama kegiatan penelitian berlangsung. Dari Gambar 3.1, terdapat beberapa tahapan yang akan dilakukan untuk menyelesaikan kasus pada penelitian tugas akhir ini yang meliputi pengumpulan data, analisa data, perancangan antar muka (*interface*), implementasi dan pengujian, dan selanjutnya kesimpulan dan saran.

3.2. Pengumpulan Data

Pengumpulan data merupakan proses yang dilakukan untuk memperoleh informasi-informasi atau data-data terhadap kasus yang menjadi permasalahan dalam laporan tugas akhir ini. Hal yang dibutuhkan penulis adalah informasi mengenai metode yang digunakan dalam penelitian kasus ini, yaitu Metode *Treemap*. Ada dua pendekatan yang penulis lakukan untuk memperoleh informasi-informasi atau pengumpulan data ini diantaranya adalah:

1. Wawancara

Wawancara merupakan metode pengumpulan data dengan mengadakan tanya jawab dan berdiskusi dalam menyelesaikan permasalahan yang dibahas dalam laporan ini dengan orang-orang yang memahami tentang kasus pembahasan mengenai integrasi data dan visualisasi menggunakan metode *Treemap*. Proses wawancara dilakukan kepada *team IT UTC GIS & DSS (IT Upstream Technical Computing Geographic Information System & Decision Support System)* PT. Chevron Pacific Indonesia.

2. Studi Pustaka (*Library Research*)

Studi pustaka dilakukan dengan tujuan untuk mengetahui metode apa yang akan digunakan dalam menyelesaikan permasalahan yang akan diteliti, serta mendapatkan dasar-dasar referensi yang kuat dalam menerapkan suatu metode yang akan digunakan dalam tugas akhir ini, yaitu dengan mempelajari buku-buku, artikel-artikel dan jurnal-jurnal yang berhubungan dengan permasalahan yang akan dibahas.

3.3. Analisa

Analisa merupakan proses yang dilakukan setelah pengumpulan terhadap data-data atau informasi mengenai kasus yang diangkat pada penelitian tugas akhir ini. Analisa merupakan proses yang digunakan untuk menganalisis masalah yaitu menganalisis data dan menganalisis mengenai metode *Treemap* yang akan digunakan.

3.4. Perancangan

Perancangan merupakan proses yang khusus dilakukan untuk merancang tampilan *form* dan menu terhadap hal yang telah dianalisa dengan tujuan untuk memberikan kemudahan dan menyederhanakan suatu proses atau jalannya metode yang diproses. Seperti perancangan antar muka sistem (*user interface*), menu-menu dan perancangan basis data yang akan digunakan.

3.5. Implementasi dan Pengujian

Proses implementasi dan pengujian yang dilakukan pada penelitian ini yaitu:

3.5.1. Implementasi

Pada proses implementasi ini akan dilakukan pembuatan modul-modul yang telah dirancang dalam tahap perancangan ke dalam bahasa pemrograman. Implementasi sistem akan dilakukan dengan spesifikasi sebagai berikut:

<i>Operating System</i>	: Windows 7 Ultimate 32 bit
<i>Memory</i>	: 4 Gb
<i>Database</i>	: Oracle 10g
<i>Tools</i>	: Java runtime 1.7

Langkah-langkah implementasi terhadap sistem yang akan dirancang adalah:

1. Menjalankan *script load data VBS* secara otomatis dengan *task scheduler operating system*, yang akan menghasilkan *CSV file*.
2. *CSV file* akan ditampilkan dalam bentuk format *Treemap*.
3. Data format *Treemap* divisualisasikan dan ditampilkan di *browser*.

4. Laporan data *Treemap* berdasarkan parameter *result*, parameter *category value*, parameter *start date* dan parameter *end date*.

3.5.2. Pengujian

Pengujian merupakan tahapan dimana aplikasi akan dijalankan. Tahap pengujian diperlukan untuk menjadi ukuran bahwa sistem dapat dijalankan sesuai dengan tujuan. Pengujian sistem dilakukan dengan menggunakan tahap pengujian *black box*. Pada *black box*, pengujian ini berfokus pada perangkat lunak untuk mendapatkan serangkaian kondisi pada sistem.

3.6. Kesimpulan dan Saran

Dalam tahap ini dapat ditentukan kesimpulan terhadap hasil pengujian yang telah dilakukan untuk mengetahui apakah implementasi sistem yang telah dilakukan dapat beroperasi dengan baik dan sesuai dengan tujuan yang diinginkan serta memberikan saran-saran untuk menyempurnakan dan mengembangkan penelitian selanjutnya.

BAB IV

ANALISA DAN PERANCANGAN

Pada perancangan perangkat lunak berbasis komputer analisa merupakan peranan yang penting dan harus diperhatikan dalam membuat rincian sistem baru. Analisa perangkat lunak merupakan langkah pemahaman persoalan sebelum mengambil tindakan atau keputusan penyelesaian hasil utama. Setelah dilaksanakan analisa, maka langkah selanjutnya adalah tahap perancangan terhadap aplikasi yang akan dibangun nantinya. Tahap perancangan sistem adalah membuat rincian sistem hasil dari analisa menjadi bentuk perancangan agar dapat dimengerti oleh pengguna (*user*).

4.1 Analisa Sistem

Berikut ini hal-hal yang termasuk dalam kategori analisa sistem yaitu:

4.1.1. Subjek Penelitian

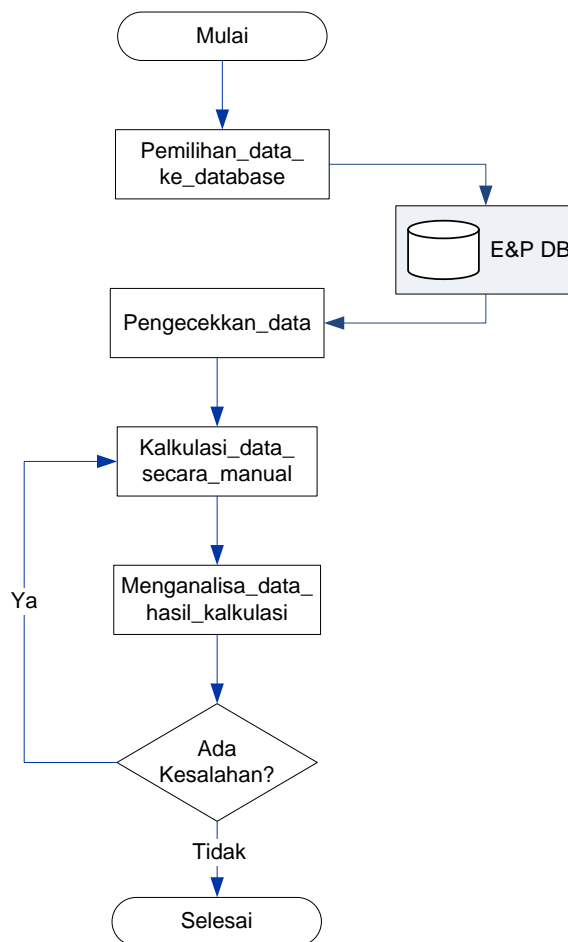
Subjek penelitian ini adalah melakukan penelitian di PT.Chevron Pacific Indonesia (PT.CPI) menggunakan metode *Treemap* yang area lokasi datanya adalah *Kalimantan Operation*.

4.1.2. Analisa Sistem Lama

Analisa terhadap sistem lama dilakukan untuk mendapatkan sebuah aplikasi yang dapat mewakili sistem yang sudah ada serta dapat mengatasi kelemahan sistem lama. Berikut ini adalah sistem lama yang digunakan PT.Chevron Pacific Indonesia dalam mengetahui suatu area yang data *production* dan data eksplorasinya kurang baik.

Data-data yang berasal dari data *exploration* dan *production* kemudian disimpan kedalam *database* oracle dalam bentuk tabel-tabel. Data disimpan berdasarkan *field-field* yang ada. Ketika seorang analis ingin melihat hasil perkembangan *well* (sumur) suatu area, baik berdasarkan tingkat hasil produksi yang *high/low* adanya *LPO (Loss Prevention Observation)* dan *on/off* suatu *well* maka analis tersebut akan mencari dan menghitung data-data tersebut secara

manual yaitu mencari max/min nya hasil produksi. Dan setiap harinya dilakukan pembaharuan data mengenai data *exploration* dan *production* tersebut. Dengan adanya data-data yang mencapai ratusan hingga ribuan ini dalam pemeriksaan data membutuhkan waktu yang lama dan *decision maker* juga tidak dapat mengambil keputusan secara cepat dan baik karena tidak memberikan hasil secara rinci tentang bagaimana keadaan suatu area *well* tersebut. *Flowchart* sistem lama dapat di gambarkan sebagai berikut:



Gambar 4. 1 *Flowchart* aktivitas sistem lama

4.1.3. Analisa Sistem Baru

Sistem baru yang akan dibangun yaitu melakukan integrasi data yang berasal dari data eksplorasi dan produksi, yang kemudian data yang telah diintegrasikan tersebut akan diproses dan dikalkulasikan sesuai kriteria masing-masing datanya, setelah data-data diproses maka data-data tersebut akan di visualisasikan dengan metode *Treemap* dengan menggunakan *browser*.

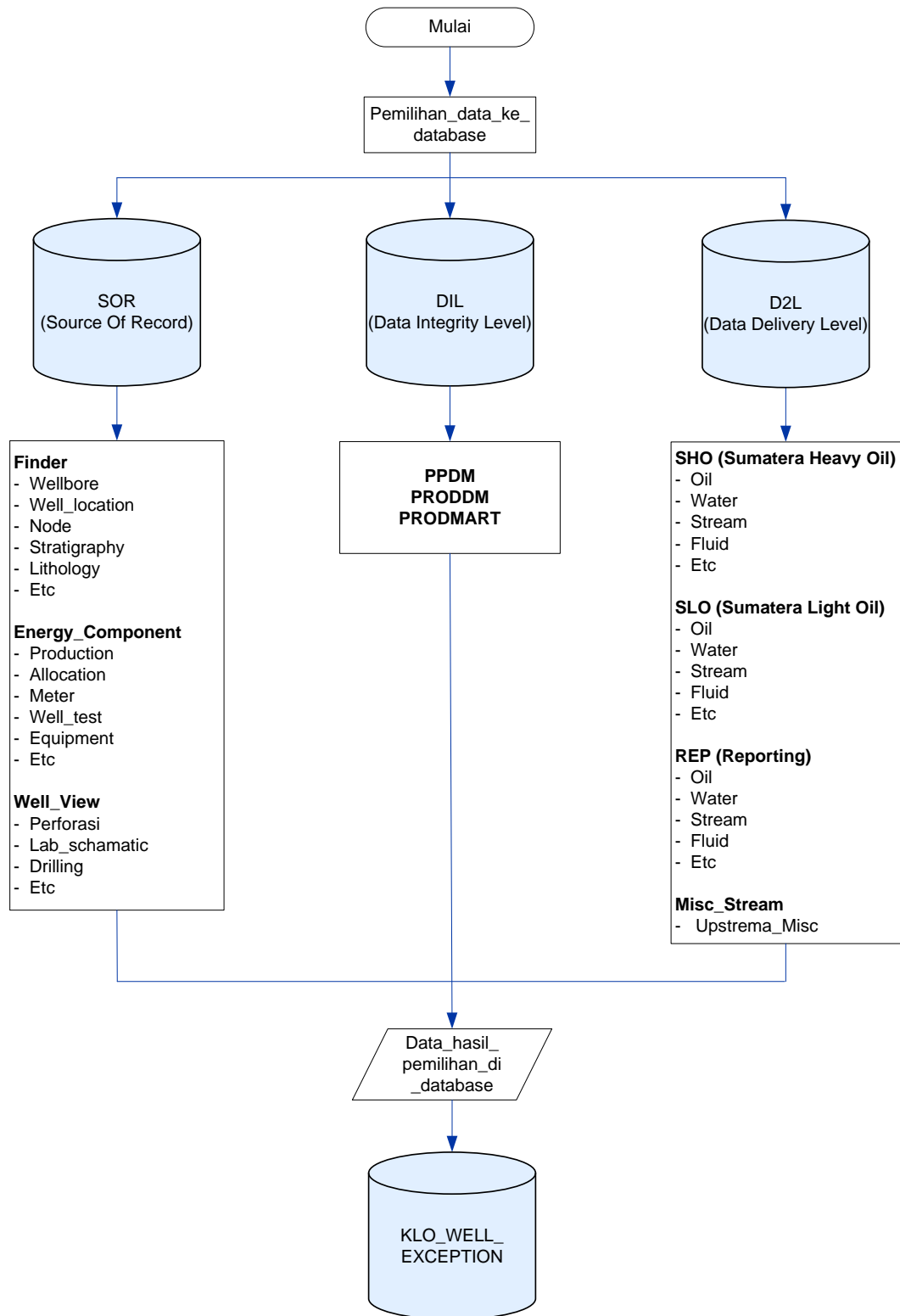
Treemap ini dapat menampilkan data dalam struktur hirarki data yang memunculkan atribut-atribut *node leaf* menggunakan ukuran dan warna. *Treemap* juga dapat menampilkan data secara cepat dan efektif. Dengan menggunakan ukuran dan warna, hal ini diharapkan dapat memudahkan *decision maker* dalam mengambil tindakan/keputusan secara cepat dan tepat dari suatu area *well*. Sistem yang akan dibangun ini juga dapat membuat laporan ke Microsoft excel dengan parameter *result*, *category value*, *start date* dan *end date*.

4.1.4. Analisa Data

Tahap analisa data dalam tahap pengerjaan analisis adalah suatu kegiatan untuk menentukan klasifikasi data untuk mendukung pembuatan rancangan basis data dengan masukan data yang lebih mudah untuk diakses dengan program aplikasi yang akan dibuat. Tahap analisa data merupakan tahapan yang sangat penting karena kesalahan di dalam tahap ini akan menyebabkan kesalahan di tahap penelitian selanjutnya.

4.1.5. Integrasi Data

Data yang akan diintegrasikan berasal dari data-data eksplorasi dan data-data produksi. *Flowchart* integrasi datanya yaitu sebagai berikut:



Gambar 4. 2 *Flowchart* integrasi data

Dari Gambar 4.2 dapat dijelaskan bahwa data-data yang akan digunakan untuk *Treemap* berasal dari tipe *record layer* SOR (*Source of Record*), DIL (*Data Integrity Level*) dan D2L (*Data Delivery Level*). SOR (*Source of Record*) terdiri dari data *Finder*, data *Energy Component* dan data *Well View*. DIL (*Data Integrity Level*) terdiri dari PPDM, PRODDM dan PRODMART, dan D2L terdiri dari SHO (*Sumatera Heavy Oil*), SLO (*Sumatera Light Oil*), REP (*Reporting*) dan *Misc Stream*. Ketiga tipe *record* ini berasal dari data produksi dan data eksplorasi. Setelah data-data diambil dan dikategorikan berdasarkan kebutuhan *Treemap*, maka akan terbentuk sebuah kategori data baru yaitu *KLO Well Exception*.

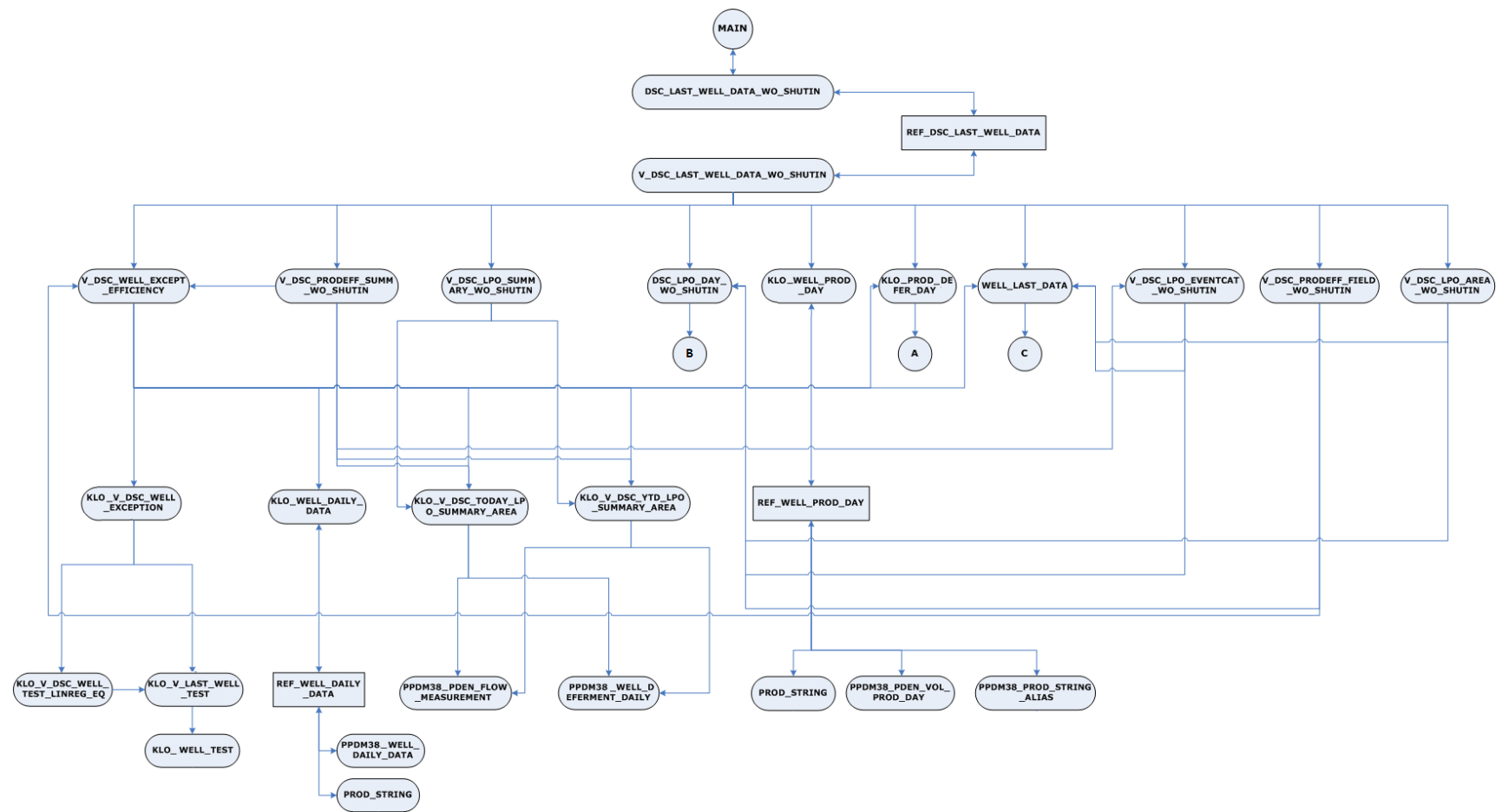
4.1.6. Analisa Kebutuhan Data

Untuk menganalisa kebutuhan data yang ada, berikut ini akan dijelaskan mengenai data-data yang di butuhkan untuk *Treemap*. Data yang dibutuhkan antara lain sebagai berikut:

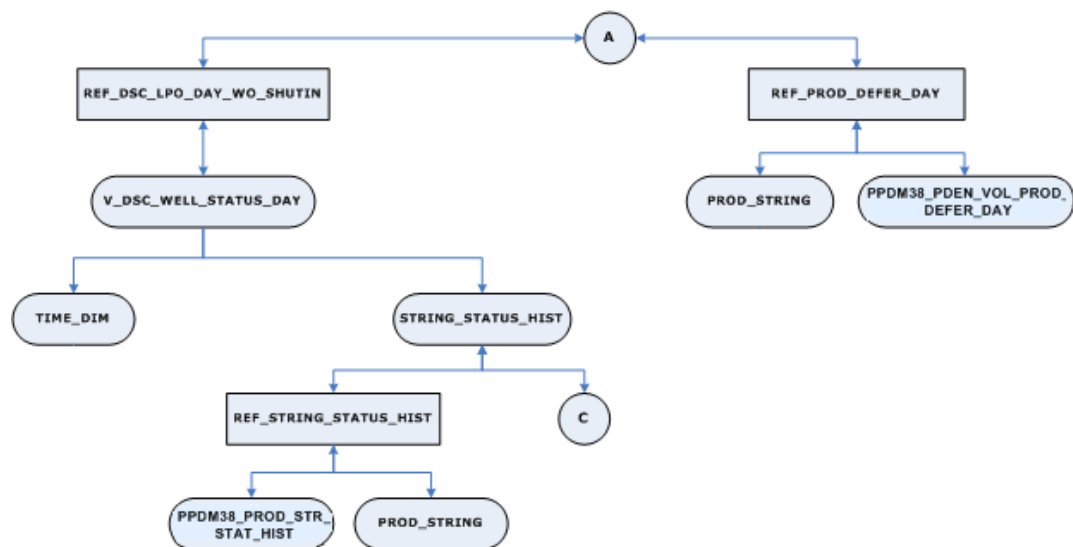
4.1.6.1. Dsc Last Well Data Wo Shutin

Dsc last well data wo shutin adalah data mengenai *well shutin* yang paling terbaru yang ada di daerah *KLO* (*Kalimantan Operation*). Dari gambar dibawah dapat dilihat proses pembentukan data yang terjadi, dimana proses data tersebut diambil dari level yang ada dibawahnya. Setelah mengalami proses integrasi data, kemudian *table*, *view* dan *procedure* tersebut akan terbentuk menjadi tabel DSC_LAST_WELL_DATA_WO_SHUTIN.

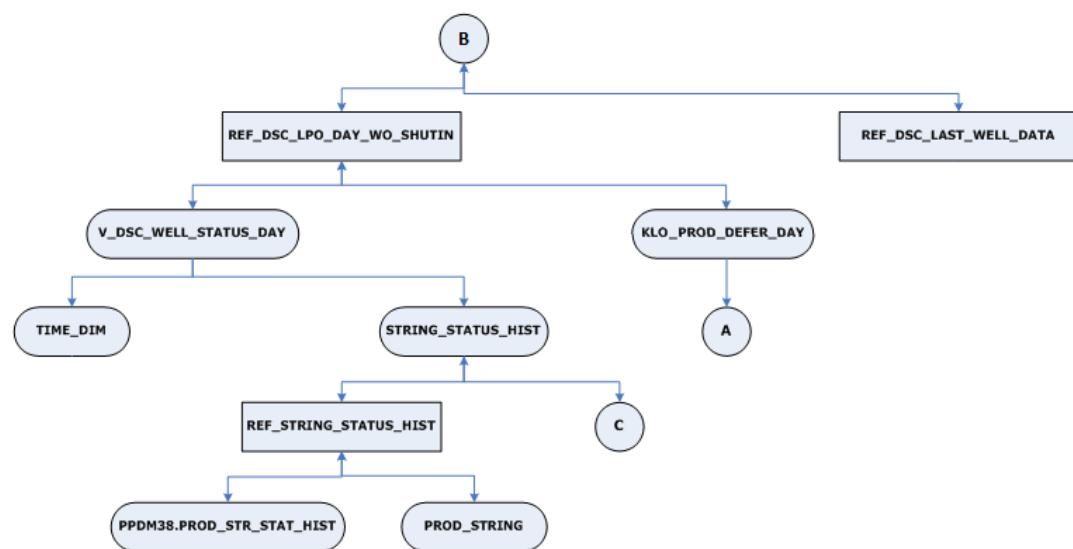
Data model dsc last well data wo shutin dapat dilihat pada Gambar 4.3, Gambar 4.4, Gambar 4.5 dan Gambar 4.6.



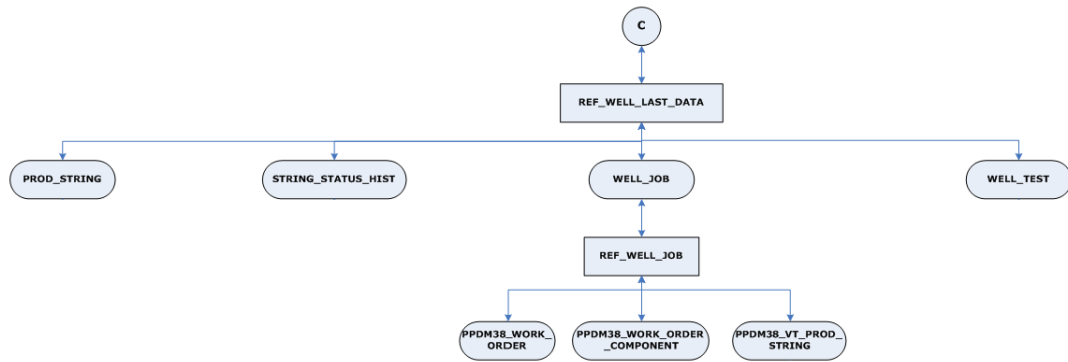
Gambar 4. 3 KLO well exception tree model – Dsc last well data wo shutin fase I



Gambar 4. 4 KLO well exception tree model – Dsc last well data wo shutin fase II



Gambar 4. 5 KLO well exception tree model – Dsc last well data wo shutin fase III



Gambar 4. 6 KLO well exception tree model – Dsc last well data wo shutin fase IV

Pada model data Gambar 4.3, Gambar 4.4, Gambar 4.5 dan Gambar 4.6 diatas, kategori dsc last well data wo shutin ini memiliki tabel-tabel antara lain :

1. DSC_LAST_WELL_DATA_WO_SHUTIN
2. DSC_LPO_DAY_WO_SHUTIN
3. KLO_WELL_PROD_DAY
4. KLO_PROD_DEFER_DAY
5. WELL_LAST_DATA
6. KLO_WELL_DAILY_DATA
7. KLO_WELL_TEST
8. PROD_STRING
9. PPDM38_WELL_DAILY_DATA
10. PPDM38_PDEN_FLOW_MEASUREMENT
11. PPDM38_WELL_DEFEREMENT_DAILY
12. PPDM38_PDEN_VOL_PROD_DAY
13. PPDM38_PROD_STRING_ALIAS
14. PPDM38_PDEN_VOL_PROD_DEFER_DAY
15. TIME_DIM
16. STRING_STATUS_HIST
17. PPDM38_PROD_STR_STAT_HIST
18. WELL_JOB
19. PPDM38_WORK_ORDER
20. PPDM38_WORK_ORDER_COMPONENT
21. PPDM38_VT_PROD_STRING

Kategori *view* dsc last well data wo shutin pada Gambar 4.3, Gambar 4.4, Gambar 4.5 dan Gambar 4.6 antara lain:

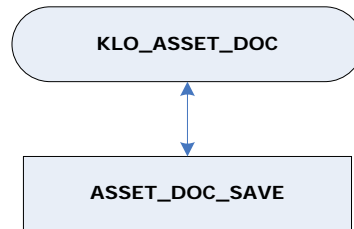
1. V_DSC_LAST_WELL_DATA_WO_SHUTIN
2. V_DSC_WELL_EXCEPT_EFFICIENCY
3. V_DSC_PRODEFF_SUMM_WO_SHUTIN
4. V_DSC_LPO_SUMMARY_WO_SHUTIN
5. V_DSC_LPO_EVENTCAT_WO_SHUTIN
6. V_DSC_PRODEFF_FIELD_WO_SHUTIN
7. V_DSC_LPO_AREA_WO_SHUTIN
8. KLO_V_DSC_WELL_EXCEPTION
9. KLO_V_DSC_TODAY_LPO_SUMMARY_AREA
10. KLO_V_DSC_YTD_LPO_SUMMARY_AREA
11. KLO_V_DSC_WELL_TEST_LINREG_EQ
12. KLO_V_LAST_WELL_TEST
13. V_DSC_WELL_STATUS_DAY

Kategori *procedure* dsc last well data wo shutin pada Gambar 4.3, Gambar 4.4, Gambar 4.5 dan Gambar 4.6 antara lain:

1. REF_WELL_PROD_DAY
2. REF_WELL_DAILY_DATA
3. REF_DSC_LPO_DAY_WO_SHUTIN
4. REF_PROD_DEFER_DAY
5. REF_STRING_STATUS_HIST
6. REF_DSC_LAST_WELL_DATA
7. REF_WELL_LAST_DATA
8. REF_WELL_JOB

4.1.6.2. KLO Asset Doc

KLO asset doc adalah data mengenai dokumen *well* yang ada di *KLO* (*Kalimantan Operation*). KLO asset doc berisi *table* dan *procedure*. Data model KLO asset doc dapat dilihat pada Gambar 4.7.

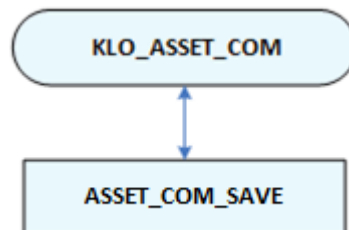


Gambar 4. 7 KLO well exception tree model – KLO asset doc

Kategori tabel KLO asset doc memiliki tabel yaitu KLO_ASSET_DOC. Kategori *procedure*-nya yaitu ASSET_DOC_SAVE.

4.1.6.3. KLO Asset Com

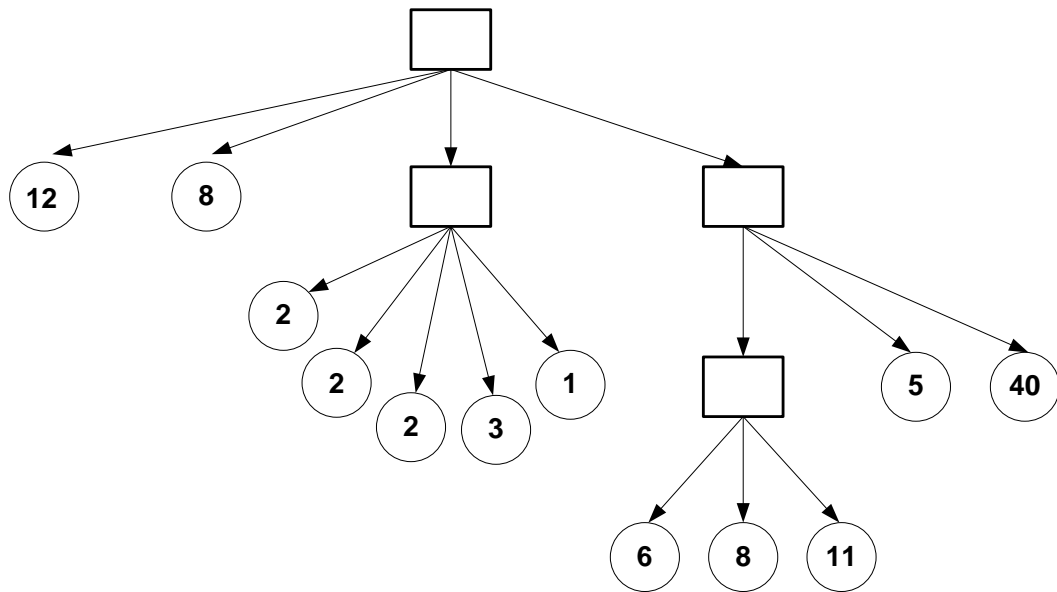
KLO asset com adalah data *remark* mengenai setiap *well* yang ada di *KLO* (*Kalimantan Operation*). KLO asset com berisi *table* dan *procedure*. Data model KLO asset com dapat dilihat pada Gambar 4.8.



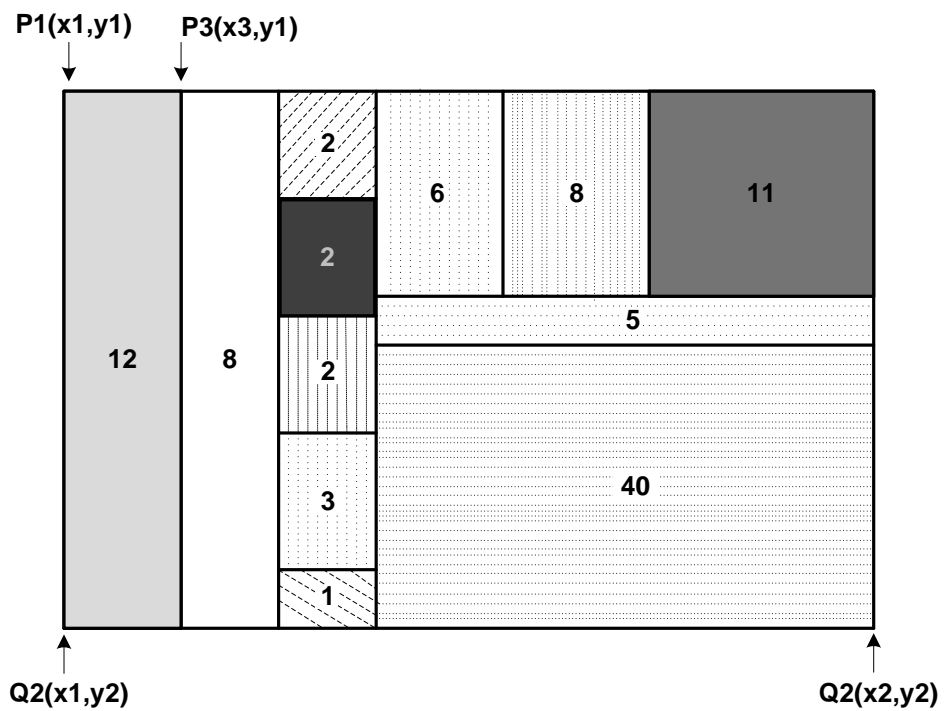
Gambar 4. 8 KLO well exception tree model – KLO asset com

Kategori tabel KLO asset com memiliki tabel yaitu KLO_ASSET_COM. Kategori *procedure*-nya yaitu ASSET_COM_SAVE.

4.1.7. Metode Treemap



Gambar 4. 9 Level struktur *tree* dengan nomor indikasi ukuran dari setiap *leaf node*



Gambar 4. 10 Implementasi *Treemap*

Metode ini membutuhkan suatu *tree root* (dapat dilihat pada gambar 4.9) dan area kotak yang didefinisikan dengan koordinat kiri atas dan kanan bawah $P1(x_1, y_1)$, $Q1(x_2, y_2)$. Jumlah sisi yang keluar dari akar menunjukkan jumlah partisi dari daerah $[x_1, y_2]$. *Subtree* paling kiri terdiri dari suatu *fraction* ($\text{Size}(\text{child}[1]) / \text{Size}(\text{root})$) dari jumlah total *bytes* pada akar (*root*).

Agar dapat dilihat dengan jelas, maka digunakan warna yang berbeda untuk tiap daerah. Efek dari melihat ribuan kotak kecil adalah seperti papan catur dengan ukuran kotak yang berbeda-beda. Pengkodean yang berwarna dapat menggambarkan besar kecilnya data, data mempunyai warna yang berbeda berdasarkan variabel warna yang telah ditentukan. Jika area yang kemudian mempunyai warna yang sama, maka sangat diperlukan garis pembatas. Karena kebutuhan pemakai sangat banyak, maka tidak ada solusi spesifik yang dapat memenuhi semua situasi itu, dan pemakai harus mempunyai suatu *control panel* untuk beberapa parameter dan juga untuk mengindikasikan warna yang mana yang digunakan untuk menandai nilai-nilai atribut (*attribute values*).

Metode *Treemap* mengasumsikan ada sebuah struktur *tree* yang mana tiap titik (*node*) itu terdiri dari *record* dengan direktorinya atau nama file (*nama*), jumlah anak (*jum_anak*), dan suatu *array of pointers* ke level berikutnya (*anak[1..jum_anak]*). Argumen atau parameter dari metode *Treemap* ini adalah:

- akar : suatu *pointer* ke akar (*root*) dari *tree* atau *subtree*.
- P,Q : *array* dengan panjang 2 dengan pasangan koordinat (x,y) dari sudut yang berseberangan dari suatu kotak tertentu (asumsikan bahwa P terdiri dari koordinat atas dan Q koordinat bawah).
- warna : menunjukkan warna yang akan digunakan untuk suatu kotak tertentu.

Sedangkan prosedur yang dibutuhkan untuk metode *Treemap* ini adalah:

- Ukuran : Suatu fungsi yang mengembalikan nilai berupa jumlah *byte* pada titik yang ditunjuk oleh parameter. Sebagai alternatif, ukuran dapat dihasilkan terlebih dahulu dan disimpan di tiap titik.

Pemanggilan awal adalah :

Treemap(akar, P, Q, warna)

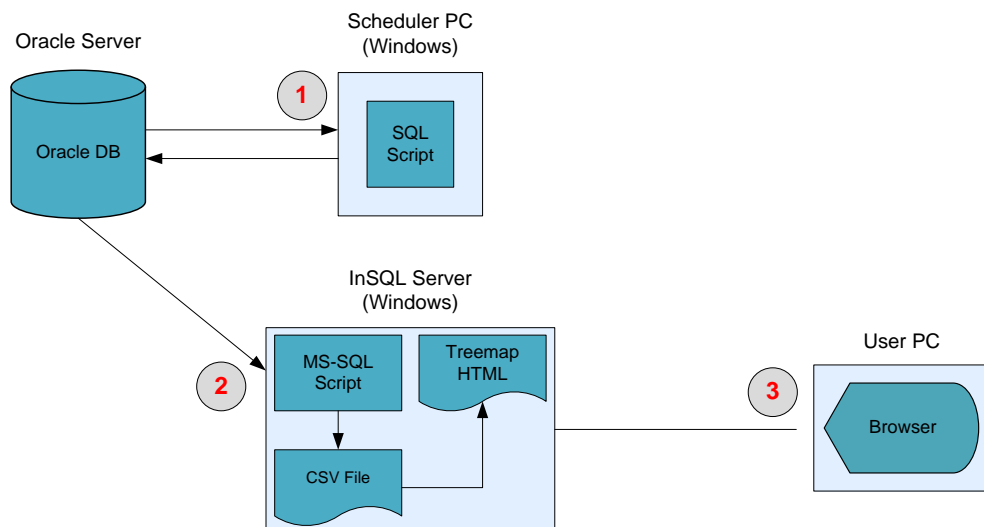
Dimana P dan Q adalah sudut kanan paling atas dan sudut kiri paling bawah dari tampilan. Dengan memberi nilai awal nol pada parameter axis, maka partisi mula-mula adalah secara vertikal. Dalam hal ini diasumsikan bahwa parameter P dan Q adalah *passed by value* :

Treemap (akar, P[0..1], Q[0..1], warna)

1. *For I := 1 to jum_anak do*
2. *Lebar := Size (anak[I]) / size(akar)) * lebar*
3. *Treemap(anak[I], P, Q, warna)*
4. *Endfor*

4.1.8. *Treemap Architecture Diagram*

Berikut ini merupakan arsitektur diagram pada *Treemap* :



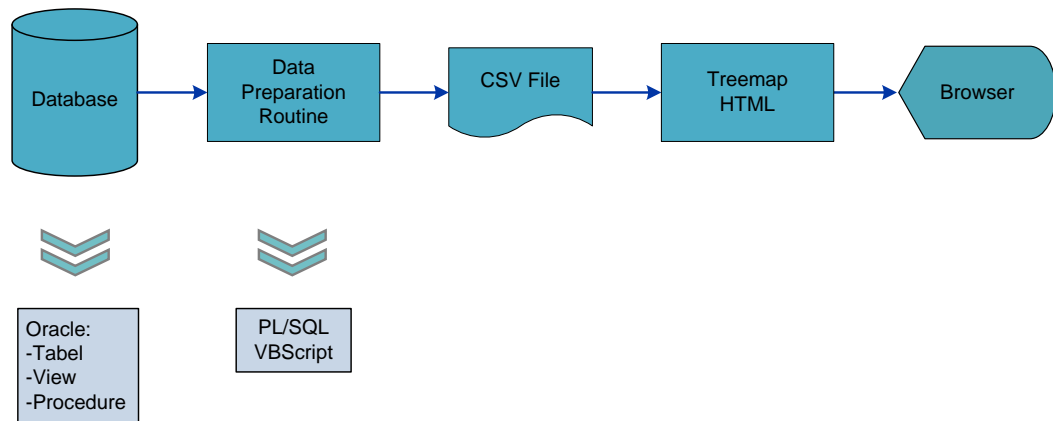
Gambar 4.11 *Treemap architecture diagram*

Pada Gambar 4.11, Oracle DB merupakan tempat penyimpanan semua *database*. Proses 1 merupakan proses untuk menjalankan *script scheduler* yang dilakukan oleh sistem operasi pada PC. Kemudian pada proses 2 akan melanjutkan proses 1 yaitu *script scheduler* yang secara otomatis akan menjalankan program yang akan di *execute* oleh komputer dengan mengambil data pada *database*, yang kemudian hasil *execute* tersebut menghasilkan file CSV. Setelah file CSV terbentuk maka akan akan divisualisasikan ke *Treemap*. Dan

pada proses 3 yaitu proses terakhir untuk menampilkan hasil dari *Treemap* di *browser*.

4.1.9. Typical Data Flow Treemap

Berikut ini merupakan langkah-langkah dalam memvisualisasikan Treemap yaitu:



Gambar 4. 12 Typical data flow treemap

1. Database

Database merupakan kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Proses *database* pada sistem ini meliputi tabel, *view* dan *procedure*.

2. Data Preparation Routine

Data Preparation Routine merupakan tempat data diproses dan dikalkulasikan sesuai kebutuhan data. Dalam fase ini menggunakan pemrograman VBScript. Berikut ini adalah *script vbs* yang digunakan.

```

Function HexToStr(ByRef strHex)
    Dim Length
    Dim Max
    Dim Str
    Max = Len(strHex)
    For Length = 1 To Max Step 2
        Str = Str & Chr("&h" & Mid(strHex, Length, 2))
    Next
    HexToStr = Str
end function
function decrypt(strencrypted)
dim strchar, ikeychar, istringchar, i
    for i = 1 to len(strencrypted)
        ikeychar = (asc(mid(g_key,i,1)))
        istringchar = asc(mid(strencrypted,i,1))
        ' *** uncomment below to decrypt with subtraction
        idecryptchar = istringchar - ikeychar
        idecryptchar = ikeychar xor istringchar
        strdecrypted = strdecrypted & chr(idecryptchar)
    next
    decrypt = strdecrypted
end function
dim db, rs, cstring
g_key = mid(readkeyfromfile("key.txt"),1,len(v_hexpwd))
v_pwd = decrypt(hextostr(v_hexpwd))
username = decrypt(hextostr("5c3655264956"))
database = decrypt(hextostr("40264a364d4f2f"))
password = v_pwd
csvexportfile="../../applet/data/KLO_well_exception_data.csv"
cstring = "provider=msdaora;data source=" & database & " ;user
    id=" & username & " ;password=" & password & ";"
set db = createobject("adodb.connection")
db.open (cstring)
query = select area, subarea, platform, field,
    dsc_last_well_data_wo_ shutin.wellnm, wellcd, mscfd,bwpd,
    bfpd, bopd, slope, lasttesdt, condition, nvl(bsw,0), gor,
    nvl(glor,0), csg_press, thp, wh_temp, boepd, well_type,
    lift_type, daytime, &_
    GLR, WOR, GL_CHOKE, SURF_CHOKE, A_MSCFD, nvl(A_BOPD, 0),
    nvl(A_BWPD, 0), PROD_BY_WELL, nvl(CSG_PRESS_D, 0), nvl(BSW_D,
    0), nvl(THP_D, 0), YTD_PROD_BY_WELL, A_BOEPD, PROD_BY_METER,
    LPO_BY_METER, YTD_PROD_BY_METER, YTD_LPO_BY_METER,
    PRODEFF_METER, YTD_PRODEFF_METER, &_
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.CreateTextFile(CSVExportFile, True)
a = rs.GetString
f.WriteLine"AREA,SUBAREA,PLATFORM,FIELD,WELLMN,WELLCD,MSCFD,BWPD
    ,BFPD,BOPD,SLOPE,LASTTESDT,CONDITION,BSW,GOR,GLOR,CSG_PRESS,TH
    P,WH_TEMP,BOEPD,WELL_TYPE,LIFT_TYPE,DAYTIME,GLR,WOR,GL_CHOKE,S
    URF_CHOKE,A_MSCFD,A_BOPD,A_BWPD,PROD_BY_WELL,CSG_PRESS_D,BSW_D
    ,THP_D,YTD_PROD_BY_WELL,A_BOEPD,PROD_BY_METER,LPO_BY_METER,YTD
    _PROD_BY_METER,YTD_LPO_BY_METER,PRODEFF_METER
f.WriteLine a
f.Close

```

3. *CSV File*

Dari data-data hasil rancangan dan kalkulasi yang ada didalam *database* kemudian di integrasikan dan diproses menggunakan VBScript dan menghasilkan file dalam bentuk CSV.

4. *Treemap*

Setelah data-data tersebut di proses dan menghasilkan file CSV kemudian data *file* CSV tersebut ditampilkan dalam bentuk format *Tremaap*.

5. *Browser*

Data format *Treemap* yang telah dihasilkan di visualisasikan dan ditampilkan di *browser* sehingga dapat berinteraksi dengan *user*.

4.1.10. Spesifikasi *Software* dan *Hardware* yang di Butuhkan

Berikut ini hal-hal yang termasuk dalam spesifikasi perangkat lunak (*software*) dan perangkat keras (*hardware*) yaitu:

4.1.10.1. *Minimal Requirement*

Berikut ini merupakan spesifikasi perangkat keras dan perangkat lunak *minimal requirement*.

Perangkat keras:

- | | |
|---------------------|------------------------------------|
| a. <i>Processor</i> | : <i>Intel Dual Core @ 1,8 Ghz</i> |
| b. <i>Memory</i> | : <i>2 GB</i> |
| c. <i>Hardisk</i> | : <i>250 GB</i> |

Perangkat Lunak:

- | | |
|--------------------------|-------------------------------|
| a. <i>Sistem Operasi</i> | : <i>Windows XP SP 3</i> |
| b. <i>DBMS</i> | : <i>Oracle 10g</i> |
| c. <i>Tools</i> | : <i>Java runtime 1.7</i> |
| d. <i>Web Server</i> | : <i>WampServer dan Xampp</i> |

4.1.10.2. Perancangan Lingkungan Pengembang

Tabel berikut ini merupakan spesifikasi perancangan perangkat keras dan perangkat lunak yang digunakan lingkungan pengembang, yaitu:

Perangkat keras:

- a. *Processor* : Intel Core i5 CPU M 460 @ 2,53 GHz
- b. *Memory* : 4 GB
- c. *Hardisk* : 500 GB
- d. *VGA/Kartu Grafik* : 1 GB

Perangkat lunak:

- a. Sistem Operasi : Windows 7 Ultimate SP 1 32-bit
- b. DBMS : Oracle 10g
- c. Bahasa Pemrograman : PL/SQL, VBScript, Java dan PHP
- d. *Tools* : Microsoft Visio 2007, StarUML, PL/SQL Developer.
- e. Web Server : WampServer

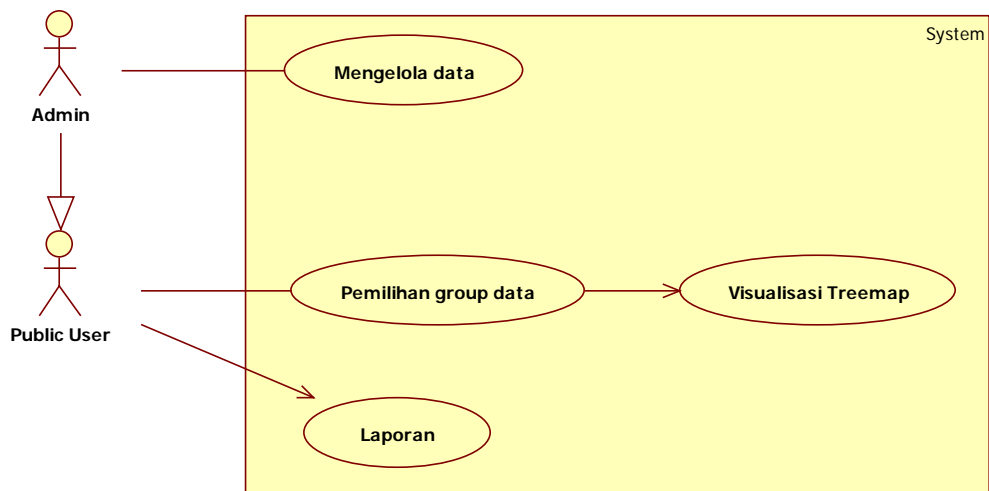
4.1.11. *Unified Modeling Language* (UML)

Perancangan aplikasi ini menggunakan *Unified Modeling Language* (UML) serta meliputi *use case diagram*, *sequence diagram*, *class diagram* dan *activity diagram*.

4.1.11.1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case diagram* sangat membantu dalam menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan *client* dan merancang *test case* untuk semua *feature* yang ada pada sistem. Pada aplikasi ini terdapat dua *actor* yaitu *admin* dan *public user*.

Adapun *use case diagram* aplikasi ini adalah sebagai berikut:



Gambar 4. 13 Use case diagram

Dari gambar 4.13 dapat dilihat sistem ini terdiri dari 2 *actor* dan 4 *use case*. Untuk lebih jelasnya, spesifikasi dari *use case diagram* dapat dilihat pada tabel dibawah ini.

Tabel 4. 1 Spesifikasi use case mengelola data

<i>Use case</i>	Mengelola data
Aktor utama	Admin
Kondisi awal	Data <i>KLO well exception</i> belum ada
Kondisi Akhir	Data <i>KLO well exception</i> sudah ada
Main success scenario	1. Admin menjalankan <i>script load data</i> untuk memasukkan data <i>KLO well exception</i> 2. <i>Script load data</i> sukses dan terbentuk <i>Csv file</i>
Alternative scenario	-

Tabel 4. 2 Spesifikasi use case pemilihan group data

<i>Use case</i>	Pemilihan <i>group data</i>
Aktor utama	Admin, <i>Public User</i>
Kondisi awal	Kriteria pemilihan data sudah ada
Kondisi Akhir	Data KLO sudah disusun berdasarkan kriteria yang dipilih
Main success scenario	1. Admin / <i>user</i> memilih menu <i>Treemap KLO</i> 2. Admin / <i>user</i> memilih indikator kriteria 3. Hasil kriteria data ditampilkan sesuai dengan yang dipilih
Alternative scenario	-

Tabel 4. 3 Spesifikasi *use case* Visualisasi *Treemap*

<i>Use case</i>	Visualisasi <i>Treemap</i>
Aktor utama	Admin, <i>Public User</i>
Kondisi awal	Visualisasi <i>Treemap</i> sudah dijalankan
Kondisi Akhir	Data berhasil ditampilkan
<i>Main success scenario</i>	<ol style="list-style-type: none"> 1. Admin / <i>user</i> memilih menu <i>Treemap KLO</i> 2. Admin / <i>user</i> memilih indikator kriteria 3. Hasil kriteria data ditampilkan sesuai dengan yang dipilih
<i>Alternative scenario</i>	-

Tabel 4. 4 Spesifikasi *use case* laporan

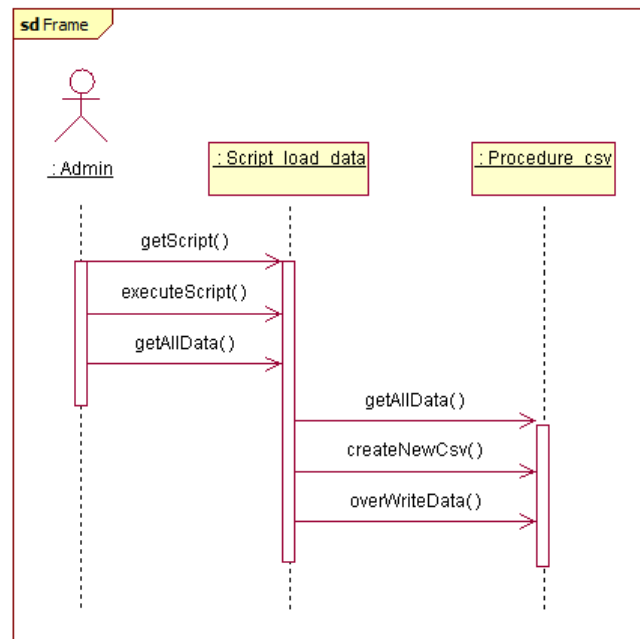
<i>Use case</i>	Laporan
Aktor utama	Admin, <i>Public User</i>
Kondisi awal	Data laporan belum ada
Kondisi Akhir	Data laporan sudah ada
<i>Main success scenario</i>	<ol style="list-style-type: none"> 1. Admin / <i>user</i> memilih menu laporan 2. Admin / <i>user</i> memilih parameter <i>result</i>, <i>category value</i>, <i>start date</i> dan <i>end date</i> 3. Laporan berhasil di <i>export</i> ke Microsoft excel
<i>Alternative scenario</i>	-

4.1.11.2. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa *message* yang digambarkan terhadap waktu.

4.1.11.2.1. *Sequence Diagram* Mengelola Data

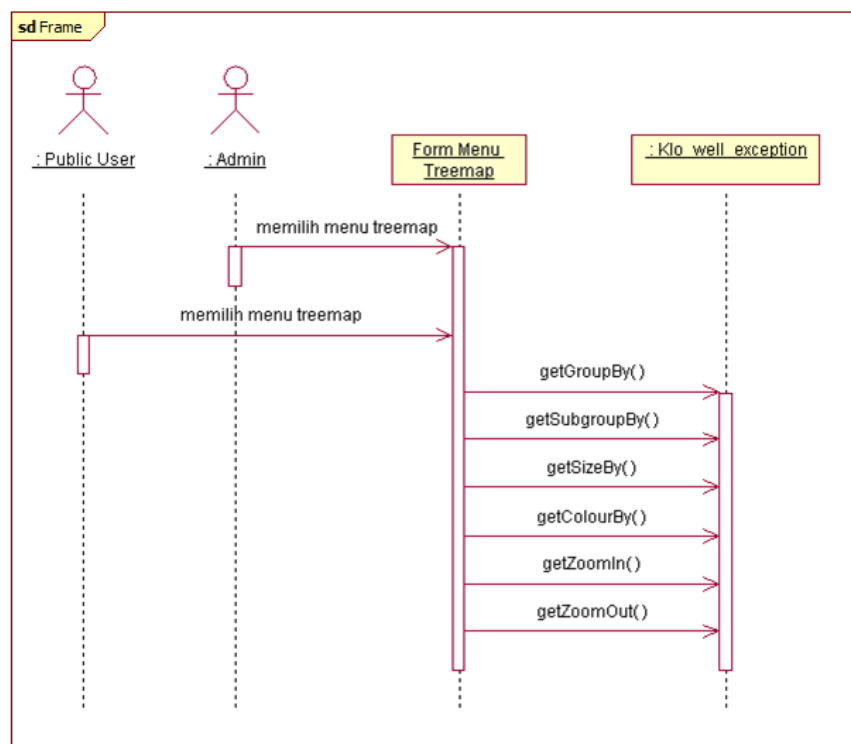
Gambar 4.14 dibawah ini menjelaskan *sequence diagram* mengelola laporan pada sistem *Treemap*.



Gambar 4. 14 *Sequence diagram* mengelola data

4.1.11.2.2. *Sequence Diagram* Visualisasi Treemap

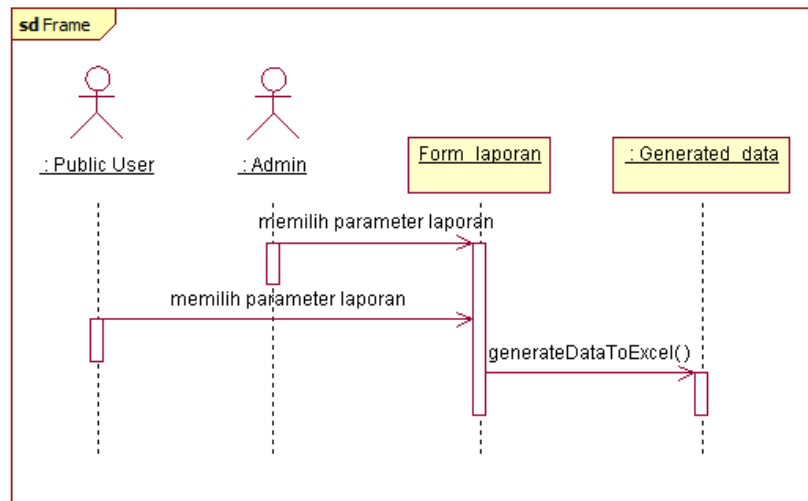
Gambar 4.15 dibawah ini menjelaskan *sequence diagram* visualisasi *Treemap* pada sistem *Treemap*.



Gambar 4. 15 *Sequence diagram* visualisasi *Treemap*

4.1.11.2.3. Sequence Diagram Menjalankan Laporan

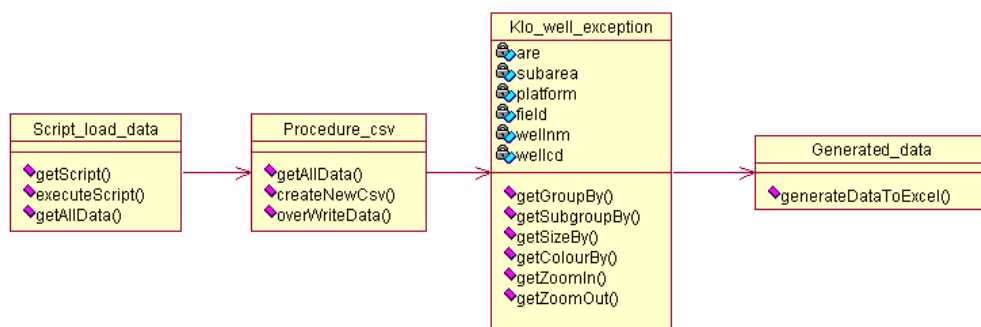
Gambar 4.16 dibawah ini menjelaskan *sequence diagram* menjalankan laporan pada sistem *Treemap*.



Gambar 4. 16 Sequence diagram menjalankan laporan

4.1.11.3. Class Diagram

Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain. *Class diagram* membantu dalam membentuk atau menentukan kelas-kelas dari suatu perangkat lunak dan kelas diagram ini merupakan tipe diagram yang paling banyak dipakai, diagram ini dapat memperlihatkan aturan-aturan dan tanggung jawab entitas-entitas yang menentukan perilaku perangkat lunak.



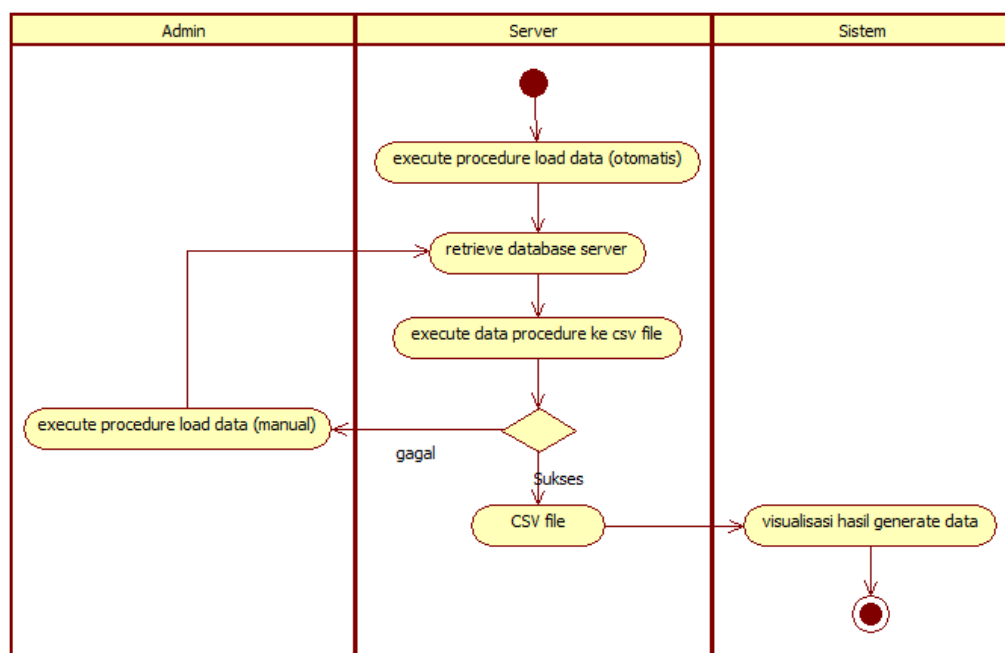
Gambar 4. 17 Class diagram

4.1.11.4. Activity Diagram

Activity diagram merupakan alur kerja pada setiap *use case*. Activity diagram menggambarkan berbagai alir aktivasi dalam sistem yang sedang dirancang. Activity diagram pada analisa ini mencakup activity diagram setiap *use case*.

4.1.11.4.1. Activity Diagram Mengelola Data

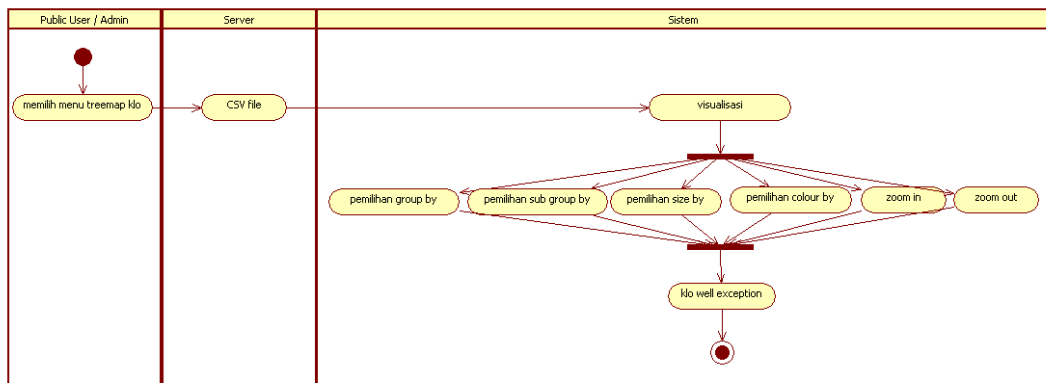
Gambar 4.18 dibawah ini menjelaskan activity diagram mengelola data pada sistem *Treemap*.



Gambar 4. 18 Activity diagram mengelola data

4.1.11.4.2. Activity Diagram Visualisasi Treemap

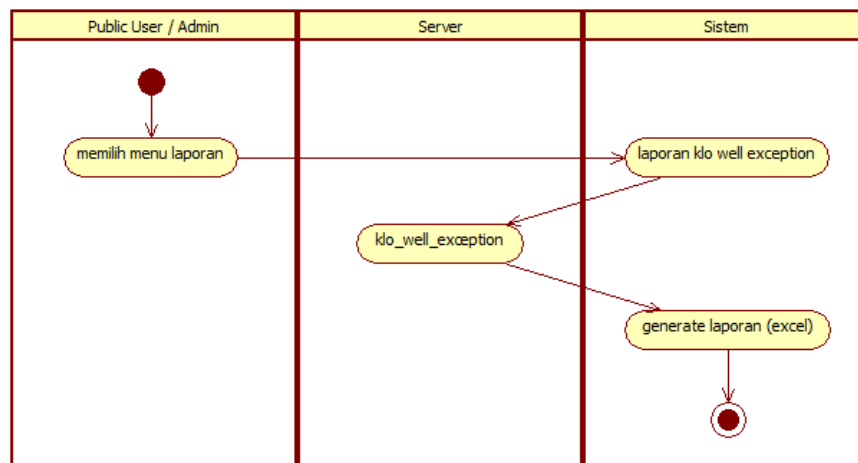
Gambar 4.19 dibawah ini menjelaskan activity diagram visualisasi *Treemap* pada sistem *Treemap*.



Gambar 4. 19 Activity diagram visualisasi Treemap

4.1.11.4.3. Activity Diagram Menjalankan Laporan

Gambar 4.20 dibawah ini menjelaskan *activity diagram* menjalankan laporan pada sistem *Treemap*.



Gambar 4. 20 Activity diagram menjalankan laporan

4.2 Perancangan Perangkat Lunak

Tujuan perancangan adalah untuk menjadi panduan pada tahap implementasi mengenai rancangan dari aplikasi yang akan dibuat dan supaya implementasi dapat dilakukan secara modular tetapi tetap konsisten.

4.2.1. Tipe Data Oracle Yang Digunakan

Tabel 4. 5 Tipe data yang digunakan di Oracle

Tipe Data	Deskripsi
VARCHAR2(n)	Data karakter dengan panjang tidak tetap. Nilai n minimum sama dengan 1 dan maksimum sama dengan 32767 byte.
NUMBER[(p,[s])]	Tipe data numerik dengan sejumlah p angka penting dan sejumlah s angka di belakang koma. Nilai p adalah <i>integer</i> positif dengan nilai maksimum 38. Nilai s dapat berada dalam rentang nilai -84 sampai 127. Nilai s <i>negative</i> berarti pembulatan sampai 10 ^s terdekat. NUMBER(p,s): bilangan <i>fixed-point</i> dengan sejumlah angka maksimum di depan koma dengan sebanyak (p-s) dan di belakang koma sebanyak s. NUMBER(p) atau number (p,0): bilangan <i>integer</i> yang mengandung angka-angka paling banyak sejumlah p. NUMBER: bilangan <i>float-point</i> dengan nilai angka penting sebanyak 38 buah, baik <i>integer</i> maupun yang mengandung pecahan.
CHAR	Menyimpan <i>string</i> /karakter dengan panjang tetap, maksimum panjangnya 2000, <i>default</i> -nya 1 byte dan akan ditempatkan di sebelah kanan sampai panjang terpenuhi dengan memakai spasi.
DATE	Data tanggal-waktu, nilai yang valid adalah mulai dari 1 Januari 4712 SM sampai 31 Desember 9999.
FLOAT	Turunan dari number presisi sampai 38 digit.

4.2.2. Entity Relationship Model (ERM)

Entity Relationship Model (ERM) atau model keterhubungan entitas merupakan model data konseptual yang mendeskripsikan hubungan antar penyimpanan. Pada model ini semesta yang ada di “Dunia Nyata” ini diterjemahkan, ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data, yang umum disebut sebagai *Entity Relationship Model (Model ER)*.

Berikut ini merupakan *Entity Relationship Model (ERM)* dari aplikasi *Treemap* yaitu sebagai berikut:

Tabel 4. 6 Keterangan ERM

No	Nama	Deskripsi	Atribut	Primary Key
1	Dsc_last_well_data_wo_shutin	Mengenai <i>well shutin</i> yang paling terbaru.	<ul style="list-style-type: none"> - Area - Subarea - Platform - Field - Wellnm - Wellcd - String_code - ... - ... - Mtd_well_lpo_control - Mtd_well_lpo_uncontrol 	<ul style="list-style-type: none"> - Wellnm - Wellcd - String_code
2	Dsc_lpo_day_wo_shutin	Mengenai data <i>well shutin</i> harian	<ul style="list-style-type: none"> - Uwi - String_id - Effective_date - Expiry_date - Daytime - End_time - String_code - ... - ... - Row_created_date - Row_quality 	<ul style="list-style-type: none"> - Uwi - String_id - String_code
3	KLO_well_prod_day	Mengenai data produksi <i>well</i> harian khusus untuk KLO	<ul style="list-style-type: none"> - Uwi - String_id - Volume_date - String_code - String_name - Alt_name - PPDM_string_code - ... - ... - Alloc_gas_lift - Theor_has_lift 	<ul style="list-style-type: none"> - Uwi - String_id - Volume_date

Penjelasan keterangan ERM (*Entity Relationship Model*) selanjutnya dapat dilihat pada lampiran A.

4.2.3. Kamus Data (*Data Dictionary*)

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data analisa sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap (Jugianto, 1999).

Perancangan kamus data yang dibutuhkan dalam membangun sistem ini adalah sebagai berikut:

1. Kamus data Dsc Last Well Data Wo Shutin

Kamus data Dsc Last Well Data Wo Shutin menjelaskan data-data *well shutin* yang paling terbaru yang ada di *Kalimantan operation*. Berikut penjelasan kamus data Dsc Last Well Data Wo Shutin yang dideskripsikan dalam tabel 4.12 di bawah ini.

Tabel 4. 7 Kamus data dsc last well data wo shutin

Nama	DSC_LAST_WELL_DATA_WO_SHUTIN
Deskripsi	Berisi data-data <i>well shutin</i> yang paling terbaru
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data <i>dsc_last_well_data_wo_shutin</i> , <i>dsc_lpo_day_wo_shutin</i> dan <i>prod_string</i> - Sebagai data <i>well shutin</i> yang paling terbaru
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	Area + subarea+ platform + ... + + <i>mtd_well_lpo_control</i> + <i>mtd_well_lpo_uncontrol</i>

2. Kamus data Dsc Lpo Day Wo Shutin

Kamus data Dsc Lpo Day Wo Shutin menjelaskan data *well shutin* harian. Berikut penjelasan kamus data Dsc Lpo Day Wo Shutin yang dideskripsikan dalam tabel 4.13 di bawah ini.

Tabel 4. 8 Kamus data dsc lpo day wo shutin

Nama	DSC_LPO_DAY_WO_SHUTIN
Deskripsi	Berisi data <i>well shutin</i> harian
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data <i>dsc_lpo_day_wo_shutin</i> dan <i>prod_string</i> - Sebagai data harian <i>well shutin</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	Uwi + String_id + Effective_date + ... + ... + Row_created_date + Row_quality

Penjelasan kamus data selanjutnya dapat dilihat pada lampiran A.

4.2.4. Perancangan Tabel

Perancangan tabel harus disesuaikan dengan kebutuhan data pada sistem. Berikut ini merupakan deskripsi tabel yang dirancang pada *database* berdasarkan ERM (gambar 4.21) di atas yaitu:

1. Tabel Dsc Last Well Data Wo Shutin

- Nama : DSC_LAST_WELL_DATA_WO_SHUTIN
- Deskripsi isi : Berisi *well shutin* yang paling terbaru
- *Primary key* : *Wellnm, wellcd, string_code*

Tabel 4. 9 Basis data dsc last well data wo shutin

Nama Field	Type dan length	Null	Default	Ket
AREA	VARCHAR2 (32)	Null	-	
SUBAREA	VARCHAR2 (9)	Null	-	
PLATFORM	VARCHAR (112)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
WELLMNM*	VARCHAR2 (30)	Not null	-	
WELLCD*	VARCHAR2 (30)	Not null	-	
STRING_CODE*	VARCHAR2 (30)	Not null		
MSCFD	NUMBER	Null	-	
BWPD	NUMBER	Null	-	
BFPD	NUMBER	Null	-	
BOPD	NUMBER	Null	-	
SLOPE	NUMBER	Null	-	
LASTTESDT	VARCHAR2 (20)	Null	-	
CONDITION	VARCHAR2 (9)	Null	-	
BSW	NUMBER	Null	-	
GOR	NUMBER	Null	-	
GLOR	NUMBER	Null	-	
CSG_PRESS	NUMBER	Null	-	
THP	NUMBER	Null	-	
WH_TEMP	NUMBER	Null	-	
BOEFD	NUMBER	Null	-	
WELL_TYPE	VARCHAR2 (20)	Null	-	
LIFT_TYPE	VARCHAR2 (20)	Null	-	
DAYTIME	DATE	Null	-	
GLR	NUMBER	Null	-	
WOR	NUMBER	Null	-	
GL_CHOKE	NUMBER	Null	-	
SURF_CHOKE	NUMBER	Null	-	
A_MSCFD	NUMBER (16,4)	Null	-	
A_BOPD	NUMBER (16,4)	Null	-	
A_BWPD	NUMBER (16,4)	Null	-	
PROD_BY_WELL	NUMBER	Null	-	
CSG_PRESS_D	NUMBER (15,6)	Null	-	
BSW_D	NUMBER (15,6)	Null	-	
THP_D	NUMBER (15,6)	Null	-	

YTD_PROD_BY_WELL	NUMBER	Null	-	
A_BOEPD	NUMBER	Null	-	
PROD_BY_METER	NUMBER	Null	-	
LPO_BY_METER	NUMBER	Null	-	
YTD_PROD_BY_METER	NUMBER	Null	-	
YTD_LPO_BY_METER	NUMBER	Null	-	
PRODEFF_METER	NUMBER	Null	-	
YTD_PRODEFF_METER	NUMBER	Null	-	
PRODEFF_KLO	NUMBER	Null	-	
YTD_PRODEFF_KLO	NUMBER	Null	-	
YTD_WELL_LPO_CONTROL	NUMBER	Null	-	
YTD_WELL_LPO_UNCONTROL	NUMBER	Null	-	
WELL_LPO_CONTROL	NUMBER	Null	-	
WELL_LPO_UNCONTROL	NUMBER	Null	-	
THEOR_FLUID	NUMBER (16,4)	Null	-	
THEOR_OIL	NUMBER (16,4)	Null	-	
THEOR_GAS	NUMBER (16,4)	Null	-	
THEOR_WTR	NUMBER (16,4)	Null	-	
THEOR_BOEPD	NUMBER	Null	-	
YTD_ALLOC_FLUID	NUMBER	Null	-	
YTD_ALLOC_OIL	NUMBER	Null	-	
YTD_ALLOC_GAS	NUMBER	Null	-	
YTD_ALLOC_WTR	NUMBER	Null	-	
YTD_ALLOC_BOEPD	NUMBER	Null	-	
YTD_THEOR_FLUID	NUMBER	Null	-	
YTD_THEOR_OIL	NUMBER	Null	-	
YTD_THEOR_GAS	NUMBER	Null	-	
YTD_THEOR_WTR	NUMBER	Null	-	
YTD_THEOR_BOEPD	NUMBER	Null	-	
HRS_OFF	NUMBER	Null	-	
FIELD_LPO_CONTROL	NUMBER	Null	-	
FIELD_LPO_UNCONTROL	NUMBER	Null	-	
YTD_FIELD_LPO_CONTROL	NUMBER	Null	-	
YTD_FIELD_LPO_UNCONTROL	NUMBER	Null	-	
PRODEFF_FIELD	NUMBER	Null	-	
YTD_PRODEFF_FIELD	NUMBER	Null	-	
WELL_STATUS	VARCHAR2 (20)	Null	-	
ON_OFF	NUMBER	Null	-	
SUBAREA_LPO_CONTROL	NUMBER	Null	-	
SUBAREA_LPO_UNCONTROL	NUMBER	Null	-	
YTD_SUBAREA_LPO_CONTROL	NUMBER	Null	-	
YTD_SUBAREA_LPO_UNCONTROL	NUMBER	Null	-	
AREA_LPO_CONTROL	NUMBER	Null	-	
AREA_LPO_UNCONTROL	NUMBER	Null	-	
YTD_AREA_LPO_CONTROL	NUMBER	Null	-	
YTD_AREA_LPO_UNCONTROL	NUMBER	Null	-	

MTD_WELL_LPO_CONTROL	NUMBER	Null	-	
MTD_WELL_LPO_UNCONTROL	NUMBER	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

2. Tabel Dsc Lpo Day Wo Shutin

- Nama : DSC_LPO_DAY_WO_SHUTIN
- Deskripsi isi : Berisi data *well shutin* harian
- *Primary key* : *Uwi, string_id, string_code*

Tabel 4. 10 Basis data dsc lpo day wo shutin

Nama Field	Type dan Length	Null	Default	Ket
UWI*	VARCHAR2 (20)	Not null	-	
STRING_ID*	VARCHAR2 (30)	Not null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
DAYTIME	DATE	Null	-	
END_TIME	DATE	Null	-	
STRING_CODE*	VARCHAR2 (30)	Not null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
ALT_NAME	VARCHAR2 (30)	Null	-	
GRID_NAME	VARCHAR2 (30)	Null	-	
PRODUCTION_DAY	DATE	Null	-	
DURATION_HOUR_SINCE_OFF	NUMBER (16,4)	Null	-	
DURATION_DAY_SINCE_OFF	NUMBER (16,4)	Null	-	
DURATION_HOUR_PER_DAY	NUMBER (16,4)	Null	-	
DEFERED_FLUID_PROD	NUMBER (16,4)	Null	-	
DEFERED_OIL_PROD	NUMBER (16,4)	Null	-	
DEFERED_WTR_PROD	NUMBER (16,4)	Null	-	
DEFERED_GAS_PROD	NUMBER (16,4)	Null	-	
PE_DEFER_OIL	NUMBER	Null	-	
OIL_BPD	NUMBER	Null	-	
AVG62_OIL_BPD	NUMBER	Null	-	
EVENT_CATAGORY	VARCHAR2 (16)	Null	-	
EVENT_REASON	VARCHAR2 (16)	Null	-	
EVENT_SYSTEM	VARCHAR2 (32)	Null	-	
SURFACE_OR_SUBSURFACE	VARCHAR2 (20)	Null	-	
PROACTIVE_OR_REACTIVE	VARCHAR2 (20)	Null	-	
EVENT_CAR_REF_TYPE	VARCHAR2 (4000)	Null	-	
REMARK	VARCHAR2 (500)	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	

OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	
ASSET_AREA	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
AREA	VARCHAR2 (32)	Null	-	
ARSE	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

Keterangan : * = *Primary Key*

** = *Foreign Key*

Perancangan tabel selanjutnya dapat dilihat pada lampiran B.

4.2.5. Perancangan View

View adalah salah satu *object database*, yang secara logika merepresentasikan sub himpunan dari data yang berasal dari satu atau lebih *table*. *View* merupakan hasil perintah *select* yang berupa tabel virtual yang dapat membungkus *query* tertentu. *View* berfungsi untuk mempermudah dalam pengambilan *query* tertentu.

1. V_DSC_WELL_EXCEPT_EFFICIENCY

Berikut ini merupakan *query view* pada V_DSC_WELL_EXCEPT_EFFICIENCY.

```
CREATE OR REPLACE VIEW V_DSC_WELL_EXCEPT_EFFICIENCY AS
select distinct A.area AREA, A.sub_area SUBAREA, A.platform
platform, a.field field, a.string_name wellnm, a.string_code
wellcd, a.mscfd mscfd, a.bwpd bwpd, a.bfpd bfpd, a.bopd bopd,
a.slope slope, a.last_test_date lasttesdt, a.condition
condition, a.bsw, a.gor, a.glor, a.csg_press, a.thp,
a.wh_temp, a.boepd, v.prod_string_type well_type,
CASE WHEN w.alloc_gas_vol IS NULL THEN w1.alloc_gas_vol ELSE
w.alloc_gas_vol END A_MSCFD,
from KLO.v_dsc_well_exception a,(select distinct v.string_name,
v.string_code, v.status, v.prod_string_type, v.production_
method from KLO.well_last_data v
where op_area_code <> 'PLGAS_OPERATION'
and status_type = 'ACT') v,
(select distinct string_name, string_code, status_date,
alloc_gas_vol, x.alloc_oil_vol, x.alloc_water_vol,
```

Query view pada V_DSC_WELL_EXCEPT_EFFICIENCY (lanjutan)

```
(nvl(x.theor_gas_vol,0) / 6) + nvl(x.theor_oil_vol,0)
total_boepd, casing_press_9, water_cut, well_head_pressure
from KLO.well_daily_data x
where x.status_date = trunc(sysdate - 1)
and op_area_code <> 'PLGAS_OPERATION'
(select distinct string_code, alloc_gas_vol,
x.alloc_oil_vol, x.alloc_water_vol,
(nvl(x.theor_gas_vol,0) / 6) + nvl(x.theor_oil_vol,0)
total_boepd
from KLO.well_daily_data x
where x.status_date = trunc(sysdate - 2)
and op_area_code <> 'PLGAS_OPERATION'
and active_ind = 'Y') w1,
(select distinct string_name, string_code, short_name,
sum((nvl(x.theor_gas_vol,0) / 6) +
nvl(x.theor_oil_vol,0)) BOEPD_Y
from KLO.well_daily_data x
where x.status_date >= trunc(sysdate, 'YEAR')
and x.status_date <= trunc(sysdate - 1)
and op_area_code <> 'PLGAS_OPERATION'
and x.active_ind = 'Y'
group by string_name, string_code, short_name) ytd_prod,
where a.string_code = v.string_code
and a.string_code = w.string_code(+)
and a.string_code = ytd_prod.string_code(+)
and a.string_code = ytd_lpo.string_code(+)
and a.SUB_AREA = sub_area_lpo.sub_area
and a.SUB_AREA = ytd_sub_area_lpo.sub_area
AND a.SUB_AREA = cnt_well.sub_area
```

Dari *query view* pada V_DSC_WELL_EXCEPT_EFFICIENCY diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel 4. 11 View v dsc well except efficiency

Nama Field	Type dan Length	Default	Ket
AREA	VARCHAR2 (32)	-	
SUBAREA	VARCHAR2 (9)	-	
PLATFORM	VARCHAR2 (112)	-	
FIELD	VARCHAR2 (32)	-	
WELLNM	VARCHAR2 (30)	-	
WELLCD	VARCHAR2 (30)	-	
MSCFD	NUMBER	-	
BWPD	NUMBER	-	
BFPD	NUMBER	-	
BOPD	NUMBER	-	
SLOPE	NUMBER	-	
LASTTESDT	VARCHAR2 (20)	-	
CONDITION	VARCHAR2 (9)	-	
BSW	NUMBER	-	
GOR	NUMBER	-	
GLOR	NUMBER	-	
CSG_PRESS	NUMBER	-	

THP	NUMBER	-	
WH_TEMP	NUMBER	-	
BOEPD	NUMBER	-	
WELL_TYPE	VARCHAR2 (20)	-	
LIFT_TYPE	VARCHAR2 (20)	-	

2. V_DSC_PRODEFF_SUMM_WO_SHUTIN

Berikut ini merupakan *query view* pada V_DSC_PRODEFF_SUMM_WO_SHUTIN.

```
CREATE OR REPLACE VIEW V_DSC_PRODEFF_SUMM_WO_SHUTIN AS
select sub_area_lpo.sub_area,
      (sub_area_lpo.prod/(sub_area_lpo.prod+ today_eventcat_lpo))*
      100 today_prodeff_meter, (ytd_sub_area_lpo.prod
      /(ytd_sub_area_lpo.prod+ ytd_eventcat_lpo)) * 100
      ytd_prodeff_meter, (by_well_lpo.prod/(by_well_lpo.prod +
      today_eventcat_lpo)) * 100 today_prodeff_well,
      (by_well_lpo.ytd_prod/(by_well_lpo.ytd_prod + ytd_eventcat_
      lpo)) * 100 ytd_prodeff_well
from
  ( select status_date, Sub_area, prod, lpo from (
    select status_date,'ATTAKA' SUB_AREA,attaka prod,
      attaka_lpo lpo from KLO.V_DSC_TODAY_LPO_SUMMARY_AREA
    union
    select status_date,'SEPINGGAN' SUB_AREA,sep prod,
      sep_lpo lpo from KLO.V_DSC_TODAY_LPO_SUMMARY_AREA
    union
    select status_date,'YAKIN' SUB_AREA,yakin prod,
      yakin_lpo lpo from KLO.V_DSC_TODAY_LPO_SUMMARY_AREA
    union
    select status_date,'WS_FPU' SUB_AREA,wss prod, wss_lpo
      lpo from KLO.V_DSC_TODAY_LPO_SUMMARY_AREA)
  ) sub_area_lpo,
  (SELECT Subarea, well_lpo_control today_eventcat_lpo,
    ytd_well_lpo_control ytd_eventcat_lpo
    from V_DSC_LPO_EVENTCAT_WO_SHUTIN) lpo
where sub_area_lpo.sub_area = ytd_sub_area_lpo.sub_area
and sub_area_lpo.sub_area = by_well_lpo.subarea
AND sub_area_lpo.sub_area = lpo.subarea
```

Dari *query view* pada V_DSC_PRODEFF_SUMM_WO_SHUTIN diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel 4. 12 View v dsc prodeff sum wo shutin

Nama Field	Type dan Length	Default	Ket
SUB_AREA	VARCHAR2 (9)	-	
TODAY_PRODEFF_METER	NUMBER	-	
YTD_PRODEFF_METER	NUMBER	-	
TODAY_PRODEFF_WELL	NUMBER	-	
YTD_PRODEFF_WELL	NUMBER	-	

Perancangan *view* selanjutnya dapat dilihat pada lampiran B.

4.2.6. Perancangan *Procedure*

Procedure adalah blok PL/SQL yang menyimpan sekumpulan perintah yang tidak disertai dengan *return value* (pengembalian nilai). *Procedure* digunakan untuk melakukan proses tertentu. *Procedure* bisa disimpan dalam *database* sebagai *object schema*, sehingga suatu *procedure* bisa digunakan berulang kali tanpa harus melakukan *parsing* dan *compile* ulang.

Berikut ini merupakan *procedure* pada REF_WELL_PROD_DAY.

```
Create or replace procedure ref_well_prod_day(ref_type
    varchar2, vd_startdate date, vd_enddate date, opu
    varchar2, vd_code varchar2 default null)
Is vc_ref_type varchar2(1);
vc_opu          varchar2(3);
vd_start_date  date;
vd_end_date    date;
pn_ref_no      number;
pn_count       number;
pn_num_i       number := 0;
pn_num_u       number := 0;
pn_proc_name   constant varchar2(60) := 'ref_well_prod_day';
pn_table_name  constant varchar2(255) := 'well_prod_day';
cursor cur_src_list (ci_opu varchar2, ci_start date, ci_end
    date) is
select uwi, string_id, string_code
from prod_string ps
where decode(vd_code, null, '1', ps.string_code) like
    decode(vd_code, null, '1', vd_code)
order by uwi, string_id;
cursor cur_src_dtl (ci_code varchar2, ci_start date, ci_end
    date) is
select ps.uwi as uwi, ps.string_id as string_id, pd.volume_date
    as volume_date, pd.pden_id as string_code, ps.string_name as
    string_name, ps.short_name as alt_name, psa.alias_code as
    PPDM_string_code, pd.alloc_fluid as alloc_fluid, pd.alloc_
    oil as alloc_oil, pd.alloc_wtr as alloc_wtr, pd.theor_oil as
    theor_oil, pd.theor_wtr as theor_wtr
From PPDM38.pden_vol_prod_day pd, prod_string ps,
    PPDM38.prod_string_alias psa
where ps.string_code = pd.pden_id and ps.uwi = psa.uwi and
    ps.string_id = psa.string_id and
    psa.alias_type='PPDM_string_code' and pd.pden_id = ci_code
and (pd.volume_date between ci_start and ci_end
    or
    trunc(coalesce(pd.row_changed_date, pd.row_created_date))
    between ci_start and ci_end+1);
if chk_argument then
    for i in a_opu.first .. a_opu.last loop
        for tab_rec in cur_src_list (a_opu(i), vd_start_date,
            vd_end_date) loop
            open cur_src_dtl(tab_rec.string_code, vd_start_date,
                vd_end_date);
```

Procedure pada REF_WELL_PROD_DAY (lanjutan)

```
        loop
            fetch cur_src_dtl into r_cur_src_dtl;
            exit when cur_src_dtl%notfound;
        end if;
        pk_log.pb_refresh('merge', vc_ref_type, pn_ref_no, sysdate,
            pn_proc_name, 'exec ('||vc_ref_type||'-'||vc_opu||'-'
            uwi:||coalesce (vd_code,'all')||')', vd_start_date, vd_
            end_date, pn_num_i, pn_num_u, pn_num_e);
        if pn_num_e = 0 and (pn_num_i > 0 or pn_num_u > 0) then
            pk_log.pb_error(pn_ref_no, pn_proc_name, 'refresh
        '||pn_table_name, 0, 'an action succeeded without warnings.');
```

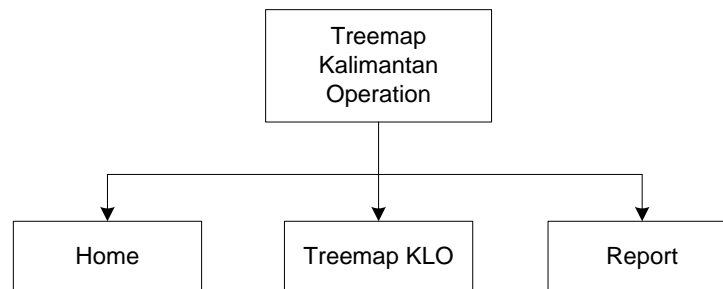
```
        elsif pn_num_e = 0 and pn_num_i = 0 and pn_num_u = 0 then
            pk_log.pb_error(pn_ref_no, pn_proc_name, 'refresh
        '||pn_table_name, 0, 'no record found.');
```

```
        end if; Commit Work; END REF_WELL_PROD_DAY;
```

4.2.7. Perancangan Antar Muka

Suatu sistem yang dirancang memerlukan antar muka (*interface*) untuk memudahkan *user* atau pengguna dalam menggunakan sistem tersebut. Perancangan antar muka suatu sistem mencakup tampilan yang baik, mudah dipahami dan tombol-tombol yang mudah dimengerti oleh pengguna.

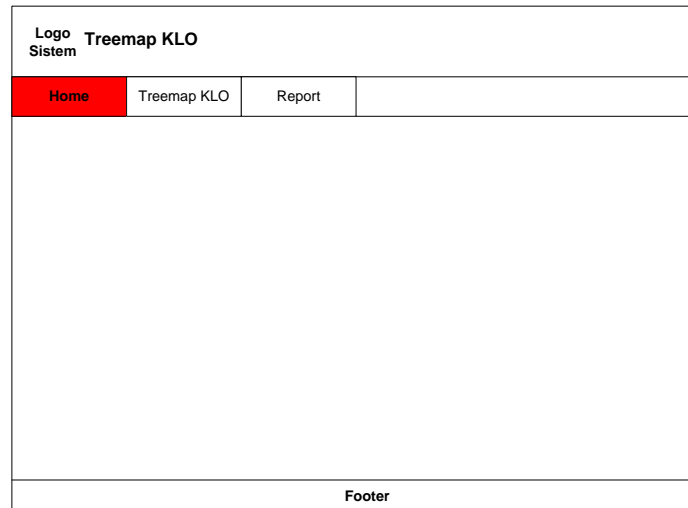
Perancangan antar muka pada bab ini hanya menampilkan *form* utama *home* (beranda), *Treemap KLO* dan *Report* (laporan). Berikut ini merupakan struktur menu dari sistem yang akan dibangun dapat dilihat pada Gambar 4.22 di bawah ini.



Gambar 4. 22 Rancangan struktur menu

4.2.7.1. Home

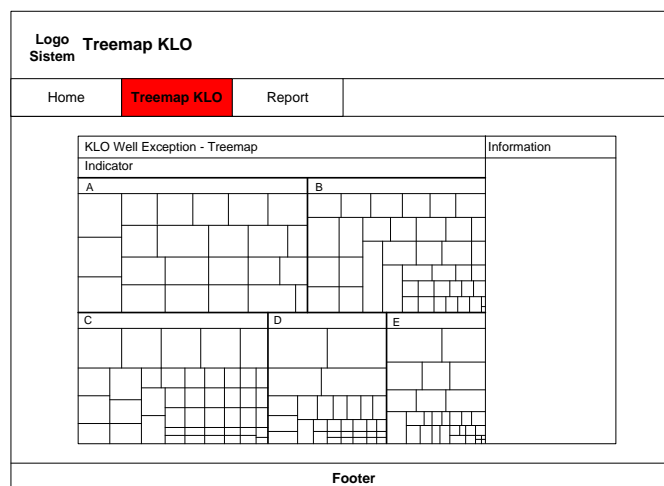
Modul *home* ini memiliki fitur informasi singkat mengenai *Treemap Kalimantan Operation*.



Gambar 4. 23 Perancangan menu *home*

4.2.7.2. Treemap KLO


Modul *Treemap KLO* ini memiliki fitur melihat data dalam bentuk *Treemap* yang disertai dengan informasi masing-masing area dan dapat mengelompokkan data *record* yang sesuai berdasarkan kategori data yaitu berdasarkan *group*, berdasarkan *sub group*, berdasarkan *size* dan berdasarkan *colour*. Kemudian *Treemap KLO* juga disertai dengan *zoom in* dan *zoom out* data.



Gambar 4. 24 Perancangan *Treemap KLO*

4.2.7.3. *Report*

Modul *Report* ini memiliki fitur pembuatan laporan berdasarkan parameter *result*, parameter *category value*, parameter *start date* dan parameter *end date* yang kemudian laporan tersebut di *export* ke Microsoft excel.

Treemap Kalimantan Operation		Logo Sistem	
Home	Treemap KLO	Report	
<div>Report</div> <div>Result : <input type="text"/></div> <div>Category Value : <input type="text"/></div> <div>Start Date : <input type="text"/> End Date : <input type="text"/></div> <div> <input type="button" value="Excel"/></div>			
Footer			

Gambar 4. 25 Perancangan menu *Report*

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi (*Implementation*)

Implementasi adalah tahap dimana sistem/aplikasi siap untuk dioperasikan pada keadaan yang sebenarnya sesuai dari hasil analisis dan perancangan yang telah dilakukan, sehingga akan diketahui apakah sistem/aplikasi yang dirancang benar-benar dapat menghasilkan tujuan yang ingin dicapai. Pada tahap ini menerapkan sistem/aplikasi ke bahasa pemrograman yang sesuai, sehingga akan diperoleh hasil yang diinginkan.

5.1.1. Batasan Implementasi

Batasan implementasi dari tugas akhir ini adalah:

1. Visualisasi *Treemap* yang digunakan yaitu berbasis 2D. Dengan visualisasi 2D dapat menghindari data bertumpuk (3D) yang dapat menyebabkan seorang *user* salah melihat data.
2. Kriteria warna sudah ditentukan dari awal saat melakukan perancangan dengan *user*.
3. Area lokasi operasi yang digunakan adalah KLO (Kalimantan Operation).
4. Data yang digunakan yaitu data eksplorasi dan data produksi.

5.1.2. Lingkungan Implementasi

Pada prinsipnya desain sistem/aplikasi yang telah dirancang memerlukan sarana pendukung yaitu berupa peralatan-peralatan (*hardware*) yang sangat berperan dalam menunjang penerapan sistem/aplikasi yang telah didesain terhadap pengolahan data. Komponen-komponen yang dibutuhkan antara lain *hardware*, yaitu kebutuhan perangkat keras komputer dalam pengolahan data kemudian *software*, yaitu kebutuhan akan perangkat lunak berupa sistem/aplikasi untuk mengoperasikan sistem/aplikasi yang telah didesain.

Perangkat lunak dan perangkat keras yang digunakan oleh PT.CPI untuk menjalankan *Treemap* adalah:

1. Perangkat Keras
 - a. *Processor* : Intel Core 2 Duo @ 3,17 GHz
 - b. *Memory* : 4 GB
 - c. *Harddisk* : 500 GB
 - d. Kartu grafik : 1 GB
2. Perangkat Lunak
 - a. Sistem operasi : Windows 7 Ultimate 32 bit
 - b. DBMS : Oracle 10g
 - c. *Tools* : Java runtime 1.7

5.1.3. Analisis Hasil

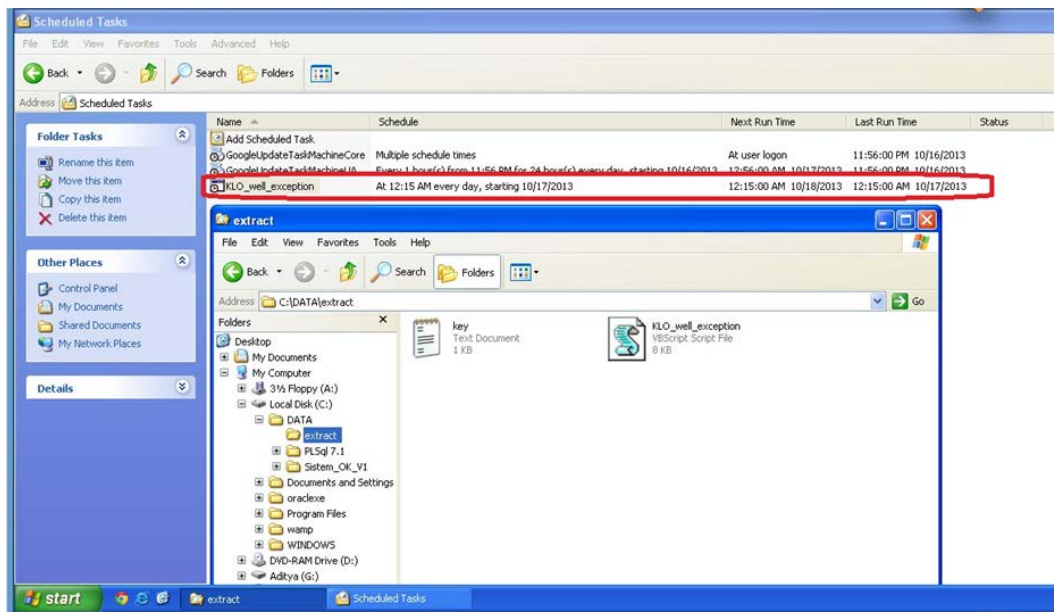
Sistem ini berbasis *desktop* yang dirancang khusus untuk *user* agar dapat mengambil keputusan secara cepat dan tepat dengan menampilkan data berdasarkan *group* data dan kategori warna dengan menyeluruh dan detail untuk setiap kondisi data *well* yang terjadi.

5.1.4. Script Load Data

Script load data dapat dilakukan dengan 2 cara yaitu *script load data* otomatis dan *script load data* manual.

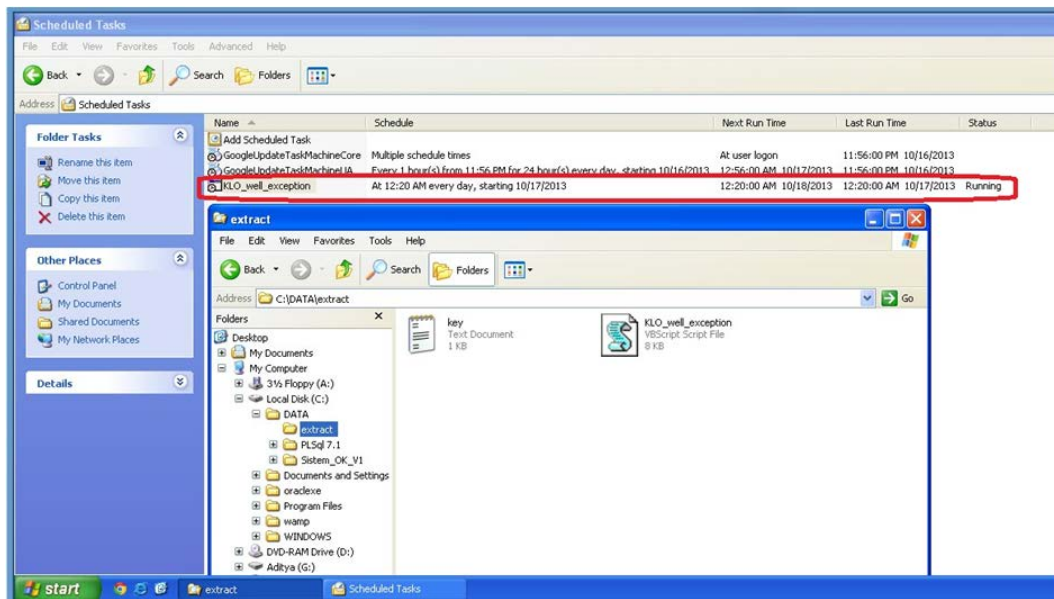
5.1.4.1. Script Load Data Otomatis

Script load data otomatis dijalankan dengan *task scheduler* sistem operasi komputer. *Script load data* dijalankan 2 kali setiap harinya. Pada gambar 5.1 dibawah ini, menjelaskan tentang *script load data* pada saat belum di *execute*.



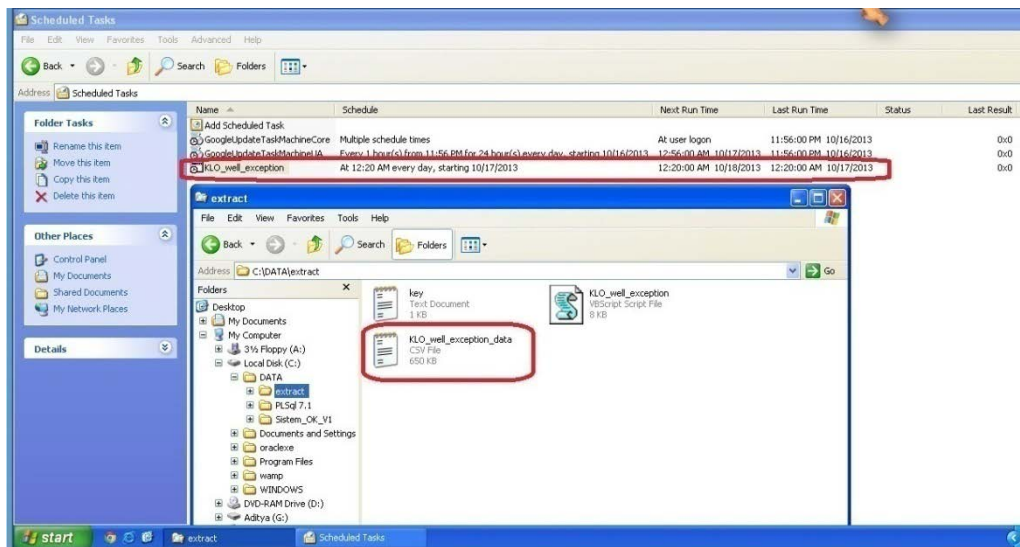
Gambar 5. 1 Script load data fase I

Pada gambar 5.2 dibawah ini, menjelaskan tentang *script load data* pada saat *running*.



Gambar 5. 2 Script load data fase II

Pada gambar 5.3 dibawah ini, menjelaskan tentang *script load data* yang telah di *execute* dan data *KLO well exception* terbentuk.



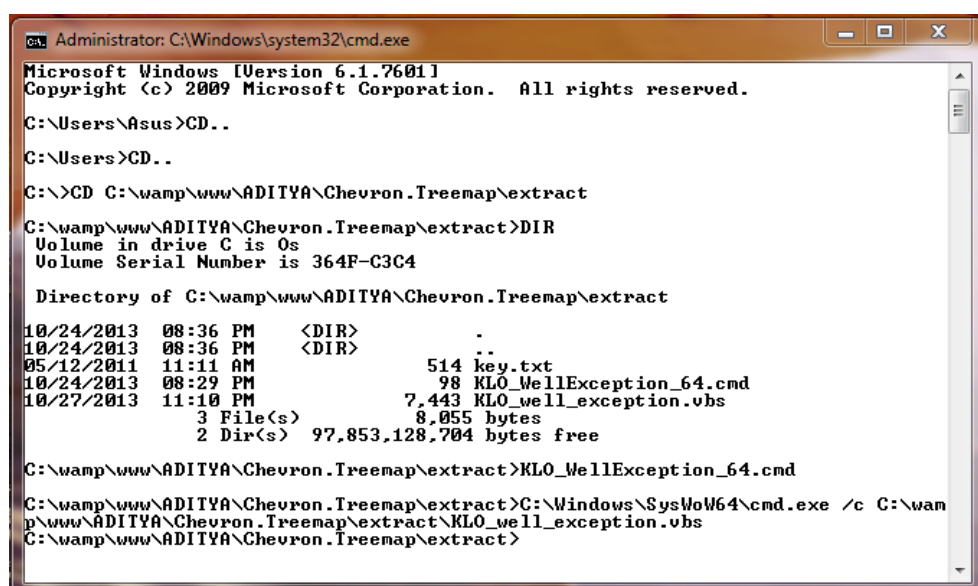
Gambar 5. 3 Script load data fase III

5.1.4.2. Script Load Data Manual

Script load data manual dijalankan apabila terjadi hal-hal berikut ini:

1. *Script load data* gagal dijalankan karena kegagalan pada sistem.
2. *Script load data* gagal dijalankan apabila data yang di *load* di sistem masih kosong, karena dilapangan terlambat menginputkannya.

Pada Gambar 5.4 dibawah ini merupakan cara menjalankan VBScript secara manual yang dilakukan oleh admin.



Gambar 5. 4 Tampilan menjalankan load script data manual

5.1.5. Implementasi Tampilan Sistem

Tahap implementasi *interface* pada sistem ini meliputi *Home*, *Treemap KLO* dan *Report*.

5.1.5.1. Tampilan Menu Utama

Tampilan menu utama ini dapat diakses oleh admin dan *user*. Tampilan menu utama sistem ini dapat dilihat pada Gambar 5.5.



Gambar 5. 5 Tampilan menu utama

5.1.5.2. Tampilan Menu *Treemap* KLO

Menu *Treemap* KLO merupakan menu untuk menampilkan data dalam bentuk *Treemap* yang disertai dengan informasi masing-masing area dan dapat mengelompokkan data *record* yang sesuai berdasarkan kategori data yaitu berdasarkan *group*, *sub group*, *size* dan *colour*. Pada menu ini *user* juga dapat *zoom in* dan *zoom out* data. Tampilan menu *Treemap* KLO dapat dilihat pada Gambar 5.6 berikut ini.



Gambar 5. 6 Tampilan menu *Treemap KLO*

5.1.5.3. Tampilan Menu *Report*

Tampilan menu *report* merupakan menu yang menampilkan pembuatan laporan berdasarkan parameter *result*, parameter *category value*, parameter *start date* dan parameter *end date* yang kemudian laporan tersebut di *export* ke Microsoft excel. Tampilan menu *report* dapat dilihat pada Gambar 5.7 berikut ini.

Gambar 5. 7 Tampilan menu *report*

Pada Gambar 5.8 berikut ini merupakan hasil *export* dari *report* yang dijalankan.

AREA	SUBAREA	PLATFORM	FIELD	WELLNM	WELLCOD	MSCFD	BWPD	BFPD	BOPD	SLOPE	LASTTESTDT	CONDITION	BSW	GOR
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-A2	P_WSA-A02	0	138	834.96	696.96	3.569528243	17-APR-2013	INCLINING	16.33	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-A3	P_WSA-A03	0	699.12	1798.92	1099.8	7.444407756	15-APR-2013	INCLINING	38.86	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-A5	P_WSA-A05ST1	0	106.22	661.39	555.17	1.256253304	11-APR-2013	INCLINING	16.06	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-B1	P_WSA-B01HZ	0	81.84	354.24	272.4	0.391739578	10-APR-2013	INCLINING	23.1	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-B4	P_WSA-B04	0	23.52	468.48	444.96	0.353885108	16-APR-2013	INCLINING	5.02	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-B7	P_WSA-B07	0	48	427.08	379.08	2.082205959	15-APR-2013	INCLINING	11.24	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-C1	P_WSA-C01	0	1168.69	2690.35	1521.66	1.633329231	07-APR-2013	INCLINING	43.44	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-C2	P_WSA-C02	307.3	196.93	868.59	671.66	5.895340868	09-MAR-2012	INCLINING	22.67	457.514472
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-D2	P_WSA-D02HZ	0	99.42	571.41	471.99	4.849048458	16-APR-2013	INCLINING	17.4	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-D5	P_WSA-D05	0	165.04	897.2	732.16	4.454869466	16-APR-2013	INCLINING	18.4	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-D6	P_WSA-D06	0	198.96	780.56	581.6	3.373964616	17-APR-2013	INCLINING	25.49	0
DEEP_WATER	WS_FPU	WS_TLP_A	WEST_SENO	WSA-D7	P_WSA-D07RD1_ERD-24	7361.45	0	73.63	73.63	7.529932767	18-APR-2013	INCLINING	99976.27119	0
SHELF_NORTH	ATTAKA	ATTAKA_A	ATTAKA	A-12SI	ATTX_A-12ST1_SI	3980	0	0	0	0.278879798	10-JUN-2008	INCLINING		
SHELF_NORTH	ATTAKA	ATTAKA_A	ATTAKA	A-6SI	ATTX_A-6RD3ST1H2_SI	487	23	96	73	4.31557377	07-SEP-2003	INCLINING	6671.232877	0
SHELF_NORTH	ATTAKA	ATTAKA_A	ATTAKA	A-9SI	ATTX_A-9_SI	0	28	308	280	0.543117444	02-JUL-2004	INCLINING		
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-11SI	ATTX_B-11RD2_SI	0	0	96	96	0.182871148	04-MAY-2010	INCLINING	0	0
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-12SI	ATTX_B-12RD2_SI	225	4244.6	4468	233.4	1.79640753	18-APR-2013	INCLINING	95	1007.163041
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-17SI	ATTX_B-17_SI	364.98	507.78	546	38.22	0.714988756	16-APR-2013	INCLINING	93	9549.450549
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-19SI	ATTX_B-19RD1H2_SI	496.99	3546.43	3772.8	226.37	3.361960077	13-APR-2013	INCLINING	94	2195.504665
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-21SI	ATTX_B-21RD1_SI	70.96	0	40	40	0.287437901	24-MAR-2013	INCLINING		1774
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-24SI	ATTX_B-24RD1_SI	100.8	0	0	0	0.003488381	29-MAR-2012	INCLINING		0
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-35SI	ATTX_B-3RD1_SI	360	855	1140	285	7.83587787	18-APR-2013	INCLINING	75	1263.157895
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-45SI	ATTX_B-4RD3H2_SI	394.98	7031.85	7770	738.15	3.143018484	15-APR-2013	INCLINING	90.5	535.094493
SHELF_NORTH	ATTAKA	ATTAKA_B	ATTAKA	B-8SI	ATTX_B-8RD1_SI	393.6	565.49	1142.4	576.91	32.15484375	17-APR-2013	INCLINING	49.5	682.253099

Gambar 5. 8 Report excel

5.2. Pengujian Sistem

Pengujian sistem dilakukan terhadap program yang telah dirancang. Pengujian sistem dilakukan dengan tujuan untuk menjamin sistem yang dibangun sesuai dengan hasil analisa dan perancangan sehingga dapat dibuat satu kesimpulan akhir.

5.3. Pengujian Sistem dengan *Black Box*

Model atau cara pengujian pada sistem ini yaitu dengan menggunakan *black box*. Pengujian sistem yang dilakukan dengan menggunakan *black box* adalah:

5.3.1. Modul Pengujian *Script Load Data*

Prekondisi :

1. Dapat dijalankan secara otomatis dengan *task scheduler* sistem operasi komputer.
2. Dapat dijalankan secara manual oleh admin.

Tabel 5. 1 Butir uji modul pengujian *script load data*

Deskripsi	Prekon-disi	Prosedur Pengujian	Masu-kan	Keluaran yang diharap-kan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesim-pulan
Pengujian <i>script load data</i>	1. Dapat dijalankan secara otomatis dengan <i>task scheduler</i> sistem operasi komputer. 2. Dapat dijalankan secara manual oleh admin.	1. <i>Execute procedure load data</i> 2. <i>Retrieve database server</i> 3. <i>Execute data procedure</i> ke csv file 4. Membuat data Csv file	-	Menghasil-kan csv file	Data yang diambil sesuai dengan yang diharap-kan	Mengha-silkan csv file	Di terima

5.3.2. Modul Pengujian Tampil *Treemap* KLO

Prekondisi :

1. Dapat dibuka dari layar menu *Treemap* KLO
2. Untuk memvisualisasikannya pilih kategori berdasarkan *group*, *subgroup*, *size* dan *color*.

Tabel 5. 2 Butir uji modul pengujian tampil *Treemap* KLO

Deskripsi	Prekon-disi	Prosedur Pengujian	Masu-kan	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesim-pulan
Pengujian tampil <i>treemap</i> KLO	Tampilan layar menu <i>treemap</i> KLO	1. Klik menu <i>treemap</i> KLO 2. Akan tampil visualisasi <i>treemap</i>	Pilih data <i>record</i> sesuai kategori, yaitu berdasarkan <i>group</i> , <i>sub group</i> , <i>size</i> dan <i>colour</i> .	Tampil hasil visualisasi berdasarkan kategori yang dipilih	Layar yang ditampilkan sesuai dengan yang diharapkan	Tampil hasil visualisasi berdasarkan kategori yang dipilih	Di terima

5.3.3. Modul Pengujian Laporan

Prekondisi :

1. Dapat dibuka dari layar menu *report*
2. Pilih parameter *result*, *category value*, *start date* dan *end date* untuk laporan yang ingin di *export* ke Microsoft excel

Tabel 5. 3 Butir uji modul pengujian laporan

Deskripsi	Prekon-disi	Prosedur Pengujian	Masu-kan	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesim-pulan
Pengujian tampil laporan	Tampilan layar menu <i>report</i>	1. Klik menu <i>report</i> 2. Akan tampil menu pilihan parameter	Pilih parameter <i>result</i> , <i>category value</i> , <i>start date</i> dan <i>end date</i>	Tampil laporan yang berhasil di <i>export</i> ke Microsoft excel	Layar yang ditampilkan sesuai dengan yang diharapkan	Tampil laporan yang berhasil di <i>export</i> ke Microsoft excel	Di terima

5.4. Kesimpulan Pengujian

Pengujian berdasarkan *black box* memberikan hasil keluaran sistem sesuai yang diharapkan yaitu *script load data* dapat berjalan dengan baik, sehingga dapat memvisualisasikan data berdasarkan *group* data dengan menyeluruh dan detail untuk setiap kondisi *well* yang terjadi. Sistem ini juga dapat meng-*export* data yang ada di *database* ke Microsoft excel berdasarkan parameter *result*, parameter *category value*, parameter *start date* dan parameter *end date*.

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan hasil integrasi data dan visualisasi menggunakan metode *Treemap* dapat disimpulkan beberapa hal, yaitu:

1. *Treemap* dapat menampung ratusan atau ribuan data *item* dalam menampilkan data yang terorganisir.
2. Dengan menggunakan *Treemap* data yang terbesar diposisikan berada di sudut kiri atas dan yang terkecil berada pada sudut kanan bawah.
3. Dengan adanya *size indicator* warna pada *Treemap*, dapat menentukan kualitas sebuah data sehingga memudahkan pengguna dalam mengambil keputusan secara cepat dan tepat.
4. Sistem ini dapat memvisualisasikan data berdasarkan *group* data dengan menyeluruh dan detail untuk setiap kondisi data *well* yang terjadi.
5. Sistem ini dapat meng-*export* data yang ada di *database* ke *Microsoft Excel* berdasarkan parameter *result*, parameter *category value*, parameter *start date* dan parameter *end date*.

6.2. Saran

Berdasarkan hasil integrasi data dan visualisasi menggunakan metode *Treemap* di PT. Chevron Pacific Indonesia, diharapkan untuk ke depannya sistem *Treemap* ini dapat diintegrasikan dengan titik koordinat sehingga dapat ditampilkan pada sistem *GIS Map*.

DAFTAR PUSTAKA

- An article by Ben Shneiderman on the use of treemaps (as a guest on www.perceptualedge.com).* “Discovering Business Intelligence Using Treemap Visualizations,” Shneiderman, Ben. 11 April, 2006.
- Binus. “*Model Waterfall*” [Online] Available www.thesis.binus.ac.id, diakses 27 Juni 2013.
- Clark, A. James. “*Treemap*” [Online] Available www.isr.umd.edu, diakses 21 Maret 2013.
- Damopolii, Reinhard. “*Administrasi Database Oracle 8i*”. Penerbit PT. Elex Media Komputindo, Jakarta. 2002.
- Department of Computer Science and Human-Computer Interaction Laboratory University of Maryland.* “Tree visualization with Tree-maps : A 2-d space-filling approach,” Shneiderman, Ben. Halaman 92-99. 18 Juni 1991.
- Dharwiyanti, Sri., Wahono, Romi S. “Pengantar Unified Modeling Language (UML)” [Online] Available <http://www.ilmukomputer.com>, diakses 01 Juli 2013.
- Eindhoven University of Technology Dept. of Mathematics and Computer Science.* “Squarified Treemaps,” Bruls, Mark., Huizing, Kees., J. Van Wijk, Jarke. 2010.
- E-Indonesia Initiative 2008 (eII2008) Konferensi dan Temu Nasional Teknologi Informasi dan Komunikasi untuk Indonesia.* “Implementasi Integrasi Data Antar Sistem Informasi Untuk Mendukung Decision Support System,” Fajarhati, Pelita., dan Sugiarto, Mugi. Mei 2008.
- Fikri, Rijalul., Adam, Fuandina A., Prakoso, Imam. “*Pemrograman Java*”. Penerbit Andi, Yogyakarta. 2005.
- Hartono, Yuafanda Kholfi. “*Partitioning pada Oracle 11g*” [Online] Available www.ilmukomputer.org, diakses 22 Maret 2013.
- IBM. “*Entity Relationship Modeling with UML*” [Online] Available www.ibm.com, diakses 18 Maret 2013.
- IBM. “*Treemap Guide*” [Online] Available www.ibm.com, diakses 22 Maret 2013.
- IEEE Pasific Visualization Symposium.* “A Self Adaptive TreeMap Based Technique for visualizing Hierarchical Data in 3D,” Chaudhuri, Abon., Shen, Han-Wei. Halaman 105-112, 2009.
- IEEE Pacific Visualization Symposium, 2008. PacificVIS '08.* “A Treemap Based Method for Rapid Layout of Large Graphs,” Muelder, Christopher., Ma, Kwan-Liu Liu. Halaman 231-238, 2008.

- IEEE Transactions on Visualization and Computer Graphics*. "Balloon Focus: a Seamless Multi-Focus+Context Method for Treemaps," Tu, Ying., Shen, Han-Wei Wei. Volume: 14, halaman 1157-1164. 2008.
- IEEE-2009 13th International Conference*. "TreemapBar: Visualizing Additional Dimensions of Data in Bar Chart," Huang, Maolin Lin., Huang, Tze-Haw., Zhang., Jia-Wan. Halaman 98-103, 2009.
- IEEE-2012 16th International Conference on Information Visualisation*. "Edge Equalized Treemaps," Kobayashi, Aimi., Misue, Kazuo., Tanaka, Jiro. Halaman 7-12, 2012.
- Jern, Mikael. "*Treemap: The Multi Dimensional eXplorers newest best friend*" [Online] Available www.ncomva.se/usermeeting, diakses 18 Maret 2013.
- Jugianto, HM. "*Analisa dan Desain Sistem Informasi*". Halaman 36-40. Penerbit Andi, Yogyakarta. 2001.
- John Wiley and Sons, Inc.* "System Analysis and Design with UML Version 2.0 – Second Edition." Dennis, Alan., Wixom, Barbara Haly., Tegarden, David. 2005.
- Jolif, Christophe. "*Treemap – Introduction*" [Online] Available <http://livedocs.dojotoolkit.org> , diakses 25 Maret 2013.
- Jurnal Informatika Universitas Kristen Petra*. "Visualisasi dan Navigasi Media Penyimpanan Disk dengan menggunakan Metode Split Screen System," Yulia., Kendengis, Edy. Volume: 3, nomor 2, halaman 80-87. November 2002.
- Kadir, Abdul. "*Dasar pemrograman Java™ 2*". Penerbit Andi, Yogyakarta. 2004.
- Kadir, Abdul. "*Pemrograman Web*". Penerbit Andi dan STMIK Pelita Nusantara, Yogyakarta. 2003.
- Komputer, Wahana. "*Pemrograman VBScript*". Penerbit Andi, Yogyakarta. 2003.
- Lenzerini, Maurizio. "*Data Integration : A Theoretical Perspective*". Halaman 233-246, 2002.
- Prima Tarigan, Edi. "*Menguasai Oracle SQL*". Penerbit PT. Elex Media Komputindo, Jakarta. 2003.
- Proc. of the 2nd International IEEE Visualization Conferences*. "Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures," Johnson, Brian., Shneiderman, Ben. Halaman 284-291. Oktober 1991.
- Rohmad. "*Install database oracle 10g di Windows XP*" [Online] Available www.rohmad.net, diakses 21 Maret 2013.
- Shneiderman, Ben., Plaisant, Catherine. "*Treemaps for space-constrained visualization of hierarchies*". 2010.
- Solichin, Achmad. "*Pemrograman Web dengan PHP dan MySQL*" [Online] Available www.informatika.unsyiah.ac.id, diakses 06 Juli 2013

- Sutanta, Edhy. *“Sistem Basis Data Konsep dan Perancangannya dalam Sistem Informasi Manajemen”*. Edisi 1. Penerbit Andi Yogyakarta, Yogyakarta. 1996.
- Sutejo, Bambang. *“Instalasi Oracle & Membuat Starter Database”* [Online] Available www.klik-oracle.com, diakses 26 Maret 2013.
- Terashima, Shinichiro. *“Treemap”* [Online] Available www.cs.sunysb.edu, diakses 20 Maret 2013.
- The Data Management Association (Dama International), USA. *“The Gama Guide to The Data Management Body of Knowledge (DAMA-DMBOK Guide)”*. Mosley, Mark., Brackett, Micheal., Earley, Susan., Henderson, Deborah. Edisi 1, halaman 16. 2009.
- The McGraw-Hill Companies, New York*. *“Software Engineering : A Practitioner’s Approach”*. Pressman, Roger S. Edisi 7, halaman 392.
- The New York Times*. “Obama's 2012 Budget Proposal: How \$3.7 Trillion is Spent,” Carter, Shan., Cox, Amanda. 2011.
- The New York Times*. “The health of the car, van, SUV, and truck market,” Cox, Amanda., Fairfield, Hannah. 2010.
- TIBCO Spotfire. *“What is Treemap”* [Online] Available www.tipcospotfire.com, diakses 18 Maret 2013.
- White paper on Treemap Visualizations for Analyzing Multi-Dimensional, Hierarchical Data Sets*, Panopticon Software. [Online] Available www.panopticon.com. 2011.
- Widianto, Eko Didik. *“Pemodelan Sistem dengan UML”* [Online] Available www.didik.blog.undip.ac.id, diakses 28 Agustus 2013.
- Widigdo, Anon K. *“Dasar Pemrograman PHP dan MySQL”* [Online] Available www.ilmukomputer.com, diakses 06 Juli 2013.
- Yuliano, Triswansyah. *“Pengenalannya PHP”* [Online] Available www.ilmukomputer.org, diakses 06 Juli 2013

LAMPIRAN A

KETERANGAN ERM

(*ENTITY RELATIONSHIP MODEL*)

DAN KAMUS DATA (*DATA DICTIONARY*)

A.1. Keterangan ERM (*Entity Relationship Model*) (Lanjutan)

Tabel A. 1 Kamus data KLO well prod day

No	Nama	Deskripsi	Atribut	Primary Key
1	KLO_prod_defer_day	Mengenai data produksi harian yang mengalami penundaan khusus untuk KLO	<ul style="list-style-type: none"> - Uwi - String_id - Effective_date - Expiry_date - Daytime - End_time - String_code - ... - ... - Row_created_date - Row_quality 	<ul style="list-style-type: none"> - Uwi - String_id - Effective_date - Daytime - String_code
2	Well_last_data	Mengenai data <i>well</i> terakhir	<ul style="list-style-type: none"> - Uwi - String_id - String_code - String_name - Alt_name - Drill_type - Md - ... - ... - Intercept - Event_cat_ref_type 	<ul style="list-style-type: none"> - Uwi - String_id - String_code
3	KLO_well_daily_data	Mengenai data <i>well</i> harian khusus untuk KLO	<ul style="list-style-type: none"> - Uwi - Source - String_id - Status_date - Active_ind - Alloc_gas_vol - Alloc_gas_vol_ouom - ... - ... - Op_fcty_1_code - Op_fcty_2_code 	<ul style="list-style-type: none"> - Uwi - String_id - Status_date

4	KLO_well_test	Mengenai data <i>well test</i> khusus untuk KLO	<ul style="list-style-type: none"> - Uwi - String_id - Test_date - Test_num - String_code - String_name - Effective_date - ... - ... - Surf_choke - Gl-choke 	<ul style="list-style-type: none"> - Uwi - String_id - Test_date - Test_num
5	Prod_string	Mengenai <i>string</i> dari semua data <i>well</i>	<ul style="list-style-type: none"> - Uwi - String_id - String_code - String_name - Short_name - Wellhdr_code - Status_type - ... - ... - Row_created_date - Row_quality 	<ul style="list-style-type: none"> - Uwi - String_id - String_code
6	PPDM38_well_daily_data	Mengenai data <i>well</i> harian	<ul style="list-style-type: none"> - Uwi - Source - String_id - Status_date - Active_ind - Alloc_gas_vol - Alloc_gas_vol_ouom - ... - ... - Avg_sand_rate - Avg_sand_rate_ouom 	<ul style="list-style-type: none"> - Uwi - Source - String_id
7	PPDM38_pden_flow_measurement	Mengenai pengukuran data <i>well</i>	<ul style="list-style-type: none"> - Pden_id - Pden_type - Pden_source - Product_type - Measurement_date - Amendment_seq_no - Active_ind - ... - ... - Row_created_date - Row_quality 	<ul style="list-style-type: none"> - Pden_id - Pden_type - Pden_source - Product_type - Measurement_date - Amendment_seq_no

8	PPDM38_well_deferment_daily	Mengenai data <i>well</i> harian yang mengalami penundaan	<ul style="list-style-type: none"> - Pden_id - Pden_type - String_name - Well_name - Uwi - String_id - Field - ... - ... - Potential_method - Work_order_id 	<ul style="list-style-type: none"> - Pden_id - Pden_type
9	PPDM38_pden_vol_pro d_day	Mengenai produksi <i>volume well</i>	<ul style="list-style-type: none"> - Pden_id - Pden_type - Pden_source - Volume_method - Activity_type - Period_type - Volume_date - Amendment_seq_no - String_code - ... - ... - Theor_gas_lift - Theor_gas_lift_ouom 	<ul style="list-style-type: none"> - Pden_id - Pden_type - Pden_source - Volume_method - Activity_type - Period_type - Volume_date - Amendment_seq_no - String_code
10	PPDM38_prod_string_alias	Mengenai <i>string</i> dari semua data <i>well</i>	<ul style="list-style-type: none"> - Uwi - Prod_string_source - String_id - Source - Alias_id - Active_ind - Alias_code - ... - ... - Row_created_Date - Row_quality 	<ul style="list-style-type: none"> - Uwi - Prod_string_source - String_id - Source - Alias_id
11	PPDM38_pden_vol_pro d_defer_day	Mengenai data <i>volume</i> produksi harian yang mengalami penundaan	<ul style="list-style-type: none"> - Pden_id - Pden_type - Pden_source - Volume_method - Activity_type - Period_type - Volume_date - Amendment_seq_no - String_code - ... - ... 	<ul style="list-style-type: none"> - Pden_id - Pden_type - Pden_source - Volume_method - Activity_type - Period_type - Volume_date - Start_date_event - Amendment_seq_no

			- Deferred_oil_pro d_corrected	- String_code
12	String_status_ hist	Menganai <i>string</i> status <i>historical</i>	- Uwi - String_id - Start_time - End_time - String_code - String_name - Alt_name - ... - ... - Row_created_ date - Row_quality	- Uwi - String_id - Start_time
13	PPDM38_prod_str_stat_ hist	Mengenai data produksi <i>string</i> <i>status</i> <i>historical</i>	- Uwi - Prod_string_ source - String_id - Status_date - Active_ind - Effective_date - End_time - ... - ... - Event_system - Production_ method	- Uwi - Prod_string_ source - String_id - Status_date
14	Well_job	Mengenai pekerjaan yang ada di <i>well</i>	- Work_order_id - Uwi - String_id - String_code - String_name - Alt_name - PPDM_string code - ... - ... - Row_created_ date Row_quality	- Work_order_ id - Uwi - String_id
15	PPDM38_work_order	Mengenai urutan pekerjaan <i>well</i>	- Work_order_id - Active_ind - Complete_date - Due_date - Effective_date - Expiry_date - Final_billing_ date - ... - ... - Well_off_date - Well_on_date	Work_order_ id

16	PPDM38_work_order_component	Mengenai komponen urutan pekerjaan <i>well</i>	<ul style="list-style-type: none"> - Work_order_id - Component_id - Active_ind - Activity_obs_no - Analysis_id - Application_id - Area_id - ... - ... - Row_created_date - Row_quality 	<ul style="list-style-type: none"> - Work_order_id - Component_id - Uwi
17	PPDM38_vt_prod_string	Mengenai <i>string</i> dari semua data <i>well</i>	<ul style="list-style-type: none"> - Uwi - String_id - String_code - String_name - Alt_name - PPDM_string_code - Status_type - ... - ... - Well_name - Timestamp 	<ul style="list-style-type: none"> - Uwi - String_id
18	KLO_asset_doc	Mengenai dokumen <i>well</i>	<ul style="list-style-type: none"> - Wellnm - Fopenperf - Fcurrcompl - Flog1 - Flog2 - Flog3 - Fmap1 - ... - ... - Pathflog1 - Pathflog2 	<ul style="list-style-type: none"> - wellnm
19	KLO_asset_com	Mengenai data <i>comment</i>	<ul style="list-style-type: none"> - Wellnm - Comment_by - Comment_date - Comment_text - Last_comment 	<ul style="list-style-type: none"> - Wellnm - Comment_by - Comment_date

A.2. Kamus Data (*Data Dictionary*) (Lanjutan)

1. Kamus data KLO_WELL_PROD_DAY

Kamus data KLO_WELL_PROD_DAY menjelaskan data produksi *well* harian khusus untuk KLO. Berikut penjelasan kamus data KLO_WELL_PROD_DAY yang dideskripsikan dalam tabel A.1 di bawah ini.

Tabel A. 2 Kamus data KLO well prod day

Nama	KLO_WELL_PROD_DAY
Deskripsi	Berisi data produksi <i>well</i> harian khusus untuk KLO
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data KLO_well_prod_day - Sebagai data produksi <i>well</i> harian
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + String_id + Volume_date + ... + ... + Alloc_gas_lift + Theor_has_lift

2. Kamus data KLO_PROD_DEFER_DAY

Kamus data KLO_PROD_DEFER_DAY menjelaskan data produksi harian yang mengalami penundaan khusus untuk KLO. Berikut penjelasan kamus data KLO_PROD_DEFER_DAY yang dideskripsikan dalam tabel A.2 di bawah ini.

Tabel A. 3 Kamus data KLO prod defer day

Nama	KLO_PROD_DEFER_DAY
Deskripsi	Berisi data produksi harian yang mengalami penundaan khusus untuk KLO
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data KLO_prod_defer_day dan prod_string - Sebagai data produksi harian yang mengalami penundaan
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	Uwi + String_id + Effective_date + ... + ... + Row_created_date + Row_quality

3. Kamus data WELL_LAST_DATA

Kamus data WELL_LAST_DATA menjelaskan mengenai data *well* terakhir. Berikut penjelasan kamus data WELL_LAST_DATA yang dideskripsikan dalam tabel A.3 di bawah ini.

Tabel A. 4 Kamus data well last data

Nama	WELL_LAST_DATA
Deskripsi	Berisi data <i>well</i> terakhir
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data <i>well_last_data</i> - Sebagai data <i>well</i> terakhir
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + String_id + String_code + ... + ... + Intercept + Event_cat_ref_type

4. Kamus data KLO_WELL_DAILY_DATA

Kamus data WELL_DAILY_DATA menjelaskan data *well* harian yang mengalami penundaan khusus untuk KLO. Berikut penjelasan kamus data KLO_WELL_DAILY_DATA yang dideskripsikan dalam tabel A.4 di bawah ini.

Tabel A. 5 Kamus data KLO well daily data

Nama	KLO_WELL_DAILY_DATA
Deskripsi	Berisi data <i>well</i> harian khusus untuk KLO
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data <i>well_daily_data</i> dan <i>prod_string</i> - Sebagai data <i>well</i> harian
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + Source + String_id + ... + ... + Op_fcty_1_code + Op_fcty_2_code

5. Kamus data KLO_WELL_TEST

Kamus data KLO_WELL_TEST menjelaskan data *well* tes yang mengalami penundaan khusus untuk KLO. Berikut penjelasan kamus data KLO_WELL_TEST yang dideskripsikan dalam tabel A.5 di bawah ini.

Tabel A. 6 Kamus data KLO well test

Nama	KLO_WELL_TEST
Deskripsi	Berisi data <i>well</i> tes khusus untuk KLO
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data <i>well_test</i> dan <i>prod_string</i> - Sebagai data <i>well</i> tes
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + String_id + Test_date + ... + ... + Surf_choke + GI-choke

6. Kamus data PROD_STRING

Kamus data PROD_STRING menjelaskan *string* dari semua data *well*. Berikut penjelasan kamus data PROD_STRING yang dideskripsikan dalam tabel A.6 di bawah ini.

Tabel A. 7 Kamus data prod string

Nama	PROD_STRING
Deskripsi	Berisi <i>string</i> dari semua data <i>well</i>
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data <i>prod_string</i> - Sebagai <i>string</i> dari semua data <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + String_id + String_code + ... + ... + Row_created_date + Row_quality

7. Kamus data PPDM38_WELL_DAILY_DATA

Kamus data PPDM38_WELL_DAILY_DATA menjelaskan data *well* harian. Berikut penjelasan kamus data PPDM38_WELL_DAILY_DATA yang dideskripsikan dalam tabel A.7 di bawah ini.

Tabel A. 8 Kamus data PPDM38 well daily data

Nama	PPDM38_WELL_DAILY_DATA
Deskripsi	Berisi data <i>well</i> harian khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_well_daily_data dan prod_string - Sebagai data <i>well</i> harian
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + Source + String_id + ... + ... + Avg_sand_rate + Avg_sand_rate_ouom

8. Kamus data PPDM38_PDEN_FLOW_MEASUREMENT

Kamus data PPDM38_PDEN_FLOW_MEASUREMENT menjelaskan pengukuran data *well*. Berikut penjelasan kamus data PPDM38_PDEN_FLOW_MEASUREMENT yang dideskripsikan dalam tabel A.8 di bawah ini.

Tabel A. 9 Kamus data PPDM38 pden flow measurement

Nama	PPDM38_PDEN_FLOW_MEASUREMENT
Deskripsi	Berisi pengukuran data <i>well</i> khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_pden_flow_measurement - Sebagai pengukuran data <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Pden_id + Pden_type + Pden_source + ... + ... + Row_created_date + Row_quality

9. Kamus data PPDM38_WELL_DEFEREMENT_DAILY

Kamus data PPDM38_WELL_DEFEREMENT_DAILY menjelaskan data well harian yang mengalami penundaan. Berikut penjelasan kamus data PPDM38_WELL_DEFEREMENT_DAILY yang dideskripsikan dalam tabel A.9 di bawah ini.

Tabel A. 10 Kamus data PPDM38 well deferement daily

Nama	PPDM38_WELL_DEFEREMENT_DAILY
Deskripsi	Berisi data well harian yang mengalami penundaan khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_well_deferement_daily dan PPDM38_pden_flow_measurement - Sebagai data produksi harian yang mengalami penundaan
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Pden_id + Pden_type + String_name + ... + ... + Potential_method + Work_order_id

10. Kamus data PPDM38_PDEN_VOL_PROD_DAY

Kamus data PPDM38_PDEN_VOL_PROD_DAY menjelaskan produksi *volume well*. Berikut penjelasan kamus data PPDM38_PDEN_VOL_PROD_DAY yang dideskripsikan dalam tabel A.10 di bawah ini.

Tabel A. 11 Kamus data PPDM38 pden vol prod day

Nama	PPDM38_PDEN_VOL_PROD_DAY
Deskripsi	Berisi produksi <i>volume well</i> khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_pden_vol_prod_day, prod_string dan PPDM38_pden_flow_measurement - Sebagai produksi <i>volume well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Pden_id + Pden_type + Pden_source + ... + ... + Theor_gas_lift + Theor_gas_lift_ouom

11. Kamus data PPDM38_PROD_STRING_ALIAS

Kamus data PPDM38_PROD_STRING_ALIAS menjelaskan *string* dari semua data *well*. Berikut penjelasan kamus data PPDM38_PROD_STRING_ALIAS yang dideskripsikan dalam tabel A.11 di bawah ini.

Tabel A. 12 Kamus data PPDM38 prod string alias

Nama	PPDM38_PROD_STRING_ALIAS
Deskripsi	Berisi <i>string</i> dari semua data <i>well</i> khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_prod_string_alias dan prod_string - Sebagai <i>string</i> dari semua data <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + Prod_string_source + String_id + ... + ... + Row_created_date + Row_quality

12. Kamus data PPDM38_PDEN_VOL_PROD_DEFER_DAY

Kamus data PPDM38_PDEN_VOL_PROD_DEFER_DAY menjelaskan data *volume* produksi harian yang mengalami penundaan. Berikut penjelasan kamus data PPDM38_PDEN_VOL_PROD_DEFER_DAY yang dideskripsikan dalam tabel A.12 di bawah ini.

Tabel A. 13 Kamus data PPDM38 pden vol prod defer day

Nama	PPDM38_PDEN_VOL_PROD_DEFER_DAY
Deskripsi	Berisi data <i>volume</i> produksi harian yang mengalami penundaan khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_pden_vol_prod_defer_day, prod_string dan PPDM38_pden_flow_measurement - Sebagai data <i>volume</i> produksi harian yang mengalami penundaan
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Pden_id + Pden_type + Pden_source + ... + ... + Event_system_name + Deferred_oil_prod_corrected

13. Kamus data TIME_DIM

Kamus data TIME_DIM menjelaskan waktu dan tanggal. Berikut penjelasan kamus data TIME_DIM yang dideskripsikan dalam tabel A.13 di bawah ini.

Tabel A. 14 Kamus data time dim

Nama	TIME_DIM
Deskripsi	Berisi waktu dan tanggal
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data time_dim - Sebagai waktu dan tanggal
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Day_of_week + Day_name + Day_of_month + ... + ... + Day_of_year + Year

14. Kamus data STRING_STATUS_HIST

Kamus data STRING_STATUS_HIST menjelaskan *string* status *historical*. Berikut penjelasan kamus data STRING_STATUS_HIST yang dideskripsikan dalam tabel A.14 di bawah ini.

Tabel A. 15 Kamus data string status hist

Nama	STRING_STATUS_HIST
Deskripsi	Berisi <i>string</i> status <i>historical</i>
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data string_status_hist dan prod_string - Sebagai <i>string</i> status <i>historical</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + String_id + Start_time + ... + ... + Row_created_date + Row_quality

15. Kamus data PPDM38_PROD_STR_STAT_HIST

Kamus data PPDM38_PROD_STR_STAT_HIST menjelaskan data produksi *string status historical*. Berikut penjelasan kamus data PPDM38_PROD_STR_STAT_HIST yang dideskripsikan dalam tabel A.15 di bawah ini.

Tabel A. 16 Kamus data PPDM38 prod str stat hist

Nama	PPDM38_PROD_STR_STAT_HIST
Deskripsi	Berisi data produksi <i>string status historical</i> khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_prod_str_stat_hist dan prod_string - Sebagai data produksi <i>string status historical</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + Prod_string_source + String_id + ... + ... + Event_system + Production_method

16. Kamus data WELL_JOB

Kamus data WELL_JOB menjelaskan pekerjaan yang ada di *well*. Berikut penjelasan kamus data WELL_JOB yang dideskripsikan dalam tabel A.16 di bawah ini.

Tabel A. 17 Kamus data well job

Nama	WELL_JOB
Deskripsi	Berisi pekerjaan yang ada di <i>well</i>
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data well_job dan prod_string - Sebagai pekerjaan yang ada di <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Work_order_id + Uwi + String_id + ... + ... + Row_created_date + Row_quality

17. Kamus data PPDM38_WORK_ORDER

Kamus data PPDM38_WORK_ORDER menjelaskan urutan pekerjaan *well*. Berikut penjelasan kamus data PPDM38_WORK_ORDER yang dideskripsikan dalam tabel A.17 di bawah ini.

Tabel A. 18 Kamus data PPDM38 work order

Nama	PPDM38_WORK_ORDER
Deskripsi	Berisi urutan pekerjaan <i>well</i> khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_work_order - Sebagai urutan pekerjaan <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Work_order_id + Active_ind + Complete_date + ... + ... + Well_off_date + Well_on_date

18. Kamus data PPDM38_WORK_ORDER_COMPONENT

Kamus data PPDM38_WORK_ORDER_COMPONENT menjelaskan komponen urutan pekerjaan *well*. Berikut penjelasan kamus data PPDM38_WORK_ORDER_COMPONENT yang dideskripsikan dalam tabel A.19 di bawah ini.

Tabel A. 19 Kamus data PPDM38 work order component

Nama	PPDM38_WORK_ORDER_COMPONENT
Deskripsi	Berisi komponen urutan pekerjaan <i>well</i> khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_work_order_component dan PPDM38_work_order - Sebagai komponen urutan pekerjaan <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Work_order_id + Component_id + Active_ind + ... + ... + Row_created_date + Row_quality

19. Kamus data PPDM38_VT_PROD_STRING

Kamus data PPDM38_VT_PROD_STRING menjelaskan string dari semua data *well*. Berikut penjelasan kamus data PPDM38_VT_PROD_STRING yang dideskripsikan dalam tabel A.19 di bawah ini.

Tabel A. 20 Kamus data PPDM38 vt prod string

Nama	PPDM38_VT_PROD_STRING
Deskripsi	Berisi string dari semua data <i>well</i> khusus untuk PPDM
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data PPDM38_vt_prod_string dan PPDM38_work_order_component - Sebagai string dari semua data <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Uwi + String_id + String_code + ... + ... + Well_name + Timestamp

20. Kamus data KLO_ASSET_DOC

Kamus data KLO_ASSET_DOC menjelaskan dokumen *well* yang ada di *Kalimantan operation*. Berikut penjelasan kamus data KLO_ASSET_DOC yang dideskripsikan dalam tabel A.20 di bawah ini.

Tabel A. 21 Kamus data KLO asset doc

Nama	KLO_ASSET_DOC
Deskripsi	Berisi dokumen <i>well</i>
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data KLO_asset_doc dan dsc_last_well_data_wo_shutin - Sebagai dokumen <i>well</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Wellnm + Fopenperf + Fcurrcompl + ... + ... + Pathflog1 + Pathflog2

21. Kamus data KLO_ASSET_COM

Kamus data KLO_ASSET_COM menjelaskan data *comment* yang ada di *Kalimantan operation*. Berikut penjelasan kamus data KLO_ASSET_COM yang dideskripsikan dalam tabel A.21 di bawah ini.

Tabel A. 22 Kamus data KLO asset com

Nama	KLO_ASSET_COM
Deskripsi	Berisi data <i>comment</i>
Bentuk data	Tabel atau <i>file</i>
Sumber / tujuan	- Berasal dari data KLO_asset_com dan dsc_last_well_data_wo_shutin - Sebagai data <i>comment</i>
Periode	Diawal pembuatan sistem
Volume	Sesuai dengan banyaknya jumlah data yang ada diperlukan
Struktur data	- Wellnm + Comment_by + Comment_date + Comment_text + Last_comment

LAMPIRAN B

PERANCANGAN TABEL DAN PERANCANGAN *VIEW*

B.1. Perancangan Tabel (Lanjutan)

1. Tabel KLO_WELL_PROD_DAY

- Nama : KLO_WELL_PROD_DAY
- Deskripsi isi : Berisi data produksi *well* harian khusus untuk KLO
- *Primary key* : *Uwi, string_id, volume_date, string_code*

Tabel B. 1 Basis data well prod day

<i>Nama Field</i>	<i>Type dan Length</i>	<i>Null</i>	<i>Default</i>	<i>Ket</i>
UWI *	VARCHAR2 (20)	Not null	-	
STRING_ID*	VARCHAR2 (30)	Not null	-	
VOLUME_DATE*	DATE	Not null	-	
STRING_CODE*	VARCHAR2 (30)	Not null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
ALT_NAME	VARCHAR2 (30)	Null	-	
PPDM_STRING_CODE	VARCHAR2 (30)	Null	-	
ALLOC_FLUID	NUMBER (16,4)	Null	-	
ALLOC_OIL	NUMBER (16,4)	Null	-	
ALLOC_WTR	NUMBER (16,4)	Null	-	
ALLOC_GAS	NUMBER (16,4)	Null	-	
THEOR_FLUID	NUMBER (16,4)	Null	-	
THEOR_OIL	NUMBER (16,4)	Null	-	
THEOR_WTR	NUMBER (16,4)	Null	-	
THEOR_GAS	NUMBER (16,4)	Null	-	
REMARK	VARCHAR2 (500)	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	
ASSET_AREA	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
AREA	VARCHAR2 (32)	Null	-	
ARSE	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	

ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	
ALLOC_GAS_LIFT	NUMBER	Null	-	
THEOR_GAS_LIFT	NUMBER	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

2. Tabel KLO_PROD_DEFER_DAY

- Nama : KLO_PROD_DEFER_DAY
- Deskripsi isi : Berisi data produksi harian yang mengalami penundaan khusus untuk KLO
- Primary key : *Uwi, string_id, effective_date, day_time, string_code*
- Foreign key : *Uwi, string_id, string_code*

Tabel B. 2 Basis data prod defer day

Nama Field	Type dan Length	Null	Default	Ket
UWI* /**	VARCHAR2 (20)	Not null	-	
STRING_ID*/**	VARCHAR2 (30)	Not null	-	
EFFECTIVE_DATE*	DATE	Not null	-	
EXPIRY_DATE	DATE	Null	-	
DAYTIME *	DATE	Not null	-	
END_TIME	DATE	Null	-	
STRING_CODE*/**	VARCHAR2 (30)	Not null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
ALT_NAME	VARCHAR2 (30)	Null	-	
GRID_NAME	VARCHAR2 (30)	Null	-	
PRODUCTION_DAY	DATE	Null	-	
DURATION_HOUR_SINCE_OFF	NUMBER (16,4)	Null	-	
DURATION_DAY_SINCE_OFF	NUMBER (16,4)	Null	-	
DURATION_HOUR_PER_DAY	NUMBER (16,4)	Null	-	
DEFERED_FLUID_PROD	NUMBER (16,4)	Null	-	
DEFERED_OIL_PROD	NUMBER (16,4)	Null	-	
DEFERED_WTR_PROD	NUMBER (16,4)	Null	-	
DEFERED_GAS_PROD	NUMBER (16,4)	Null	-	
DEFERED_OIL_PROD_CORRECTED	NUMBER	Null	-	
PE_DEFER_OIL	NUMBER	Null	-	
OIL_BPD	NUMBER	Null	-	
AVG62_OIL_BPD	NUMBER	Null	-	
EVENT_CATEGORY	VARCHAR2 (16)	Null	-	
EVENT_REASON	VARCHAR2 (16)	Null	-	
EVENT_SYSTEM	VARCHAR2 (32)	Null	-	
POTENTIAL_METHOD	VARCHAR2(32)	Null	-	
SURFACE_OR_SUBSURFACE	VARCHAR2 (20)	Null	-	
PROACTIVE_OR_REACTIVE	VARCHAR2 (20)	Null	-	
REMARK	VARCHAR2 (500)	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	

OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	
ASSET_AREA	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
AREA	VARCHAR2 (32)	Null	-	
ARSE	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

3. Tabel WELL_LAST_DATA

- Nama : WELL_LAST_DATA
- Deskripsi isi : Berisi data data *well* terakhir
- *Primary key* : *Uwi, string_id, string_code*

Tabel B. 3 Basis data well last data

<i>Nama Field</i>	<i>Type dan Length</i>	<i>Null</i>	<i>Default</i>	<i>Ket</i>
UWI*	VARCHAR2 (20)	Not null	-	
STRING_ID*	VARCHAR2 (20)	Not null	-	
STRING_CODE*	VARCHAR2 (30)	Not null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
ALT_NAME	VARCHAR2 (30)	Null	-	
DRILL_TYPE	VARCHAR2 (20)	Null	-	
MD	NUMBER	Null	-	
TVD	NUMBER	Null	-	
PLUGBACK_DEPTH	NUMBER	Null	-	
PIS_DATE	DATE	Null	-	
STATUS_START_DATE	DATE	Null	-	
STATUS_END_DATE	DATE	Null	-	
STATUS	VARCHAR2 (20)	Null	-	
STATUS_TYPE	VARCHAR2 (20)	Null	-	
PROD_STRING_TYPE	VARCHAR2 (20)	Null	-	
PRODUCTION_METHOD	VARCHAR2 (20)	Null	-	
EVENT_CATEGORY	VARCHAR2 (16)	Null	-	
EVENT_REASON	VARCHAR2 (16)	Null	-	
EVENT_SYSTEM	VARCHAR2 (16)	Null	-	
PROACTIVE_OR_REACTIVE	VARCHAR2 (20)	Null	-	
SURFACE_OR_SUBSURFACE	VARCHAR2 (20)	Null	-	
JOB_START_DATE	DATE	Null	-	
JOB_END_DATE	DATE	Null	-	

JOB_NUMBER	VARCHAR2 (30)	Null	-	
JOB_STATUS	VARCHAR2 (50)	Null	-	
JOB_CATEGORY	VARCHAR2 (50)	Null	-	
JOB_TYPE	VARCHAR2 (50)	Null	-	
JOB_DETAIL	VARCHAR2 (50)	Null	-	
JOB_DETAIL_DESCRIPTION	VARCHAR2 (50)	Null	-	
JOB_REMARK	VARCHAR2 (5000)	Null	-	
TARGET_OIL	NUMBER	Null	-	
ESTIMATE_COST	NUMBER	Null	-	
ACTUAL_COST	NUMBER	Null	-	
TEST_DATE	DATE	Null	-	
TEST_TIME	VARCHAR2 (20)	Null	-	
ZONE_NAME	VARCHAR2 (32)	Null	-	
SAND_NAME	VARCHAR2 (255)	Null	-	
PLATFORM	VARCHAR2 (32)	Null	-	
BSW_PERCENT	NUMBER	Null	-	
WH_CSG_PRESS	NUMBER	Null	-	
WH_PRESS	NUMBER	Null	-	
WH_TEMP	NUMBER	Null	-	
OIL_RATE	NUMBER	Null	-	
OIL_GRAVITY	NUMBER	Null	-	
WATER_CUT	NUMBER	Null	-	
WATER_RATE	NUMBER	Null	-	
GROSS_FLUID	NUMBER	Null	-	
SURF_CHOKE	NUMBER	Null	-	
GL_CHOKE	NUMBER	Null	-	
SEP_PRESS	NUMBER	Null	-	
SAND_RATE	NUMBER	Null	-	
EMULSION	NUMBER	Null	-	
GAS_GRAVITY	NUMBER	Null	-	
GL_RATE	NUMBER	Null	-	
FM_GAS	NUMBER	Null	-	
TOTAL_GAS	NUMBER	Null	-	
GOR	NUMBER	Null	-	
GLR	NUMBER	Null	-	
GLOR	NUMBER	Null	-	
Z_FACTOR	NUMBER	Null	-	
SHRINKAGE_FACTOR	NUMBER	Null	-	
NET_OIL_CORRECTED	NUMBER	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	
ASSET_AREA	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
AREA	VARCHAR2 (32)	Null	-	
ARSE	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	

OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	
SLOPE	NUMBER	Null	-	
INTERCEPT	NUMBER	Null	-	
EVENT_CAT_REF_TYPE	VARCHAR2 (50)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

4. Tabel KLO_WELL_DAILY_DATA

- Nama : KLO_WELL_DAILY_DATA
- Deskripsi isi : Berisi data *well* harian khusus untuk KLO
- *Primary key* : *Uwi, string_id, status_date*
- *Foreign key* : *Uwi, string_id*

Tabel B. 4 Basis data well daily data

Nama Field	Type dan Length	Null	Default	Ket
UWI*/**	VARCHAR2 (20)	Not null	-	
SOURCE	VARCHAR2 (20)	Not null	-	
STRING_ID*/**	VARCHAR2 (20)	Not null	-	
STATUS_DATE*	DATE	Not null	-	
ACTIVE_IND	CHAR (1)	Null	-	
ALLOC_GAS_VOL	NUMBER (15,6)	Null	-	
ALLOC_GAS_VOL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_GL_VOL	NUMBER (15,6)	Null	-	
ALLOC_GL_VOL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_OIL_VOL	NUMBER (15,6)	Null	-	
ALLOC_OIL_VOL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_WATER_VOL	NUMBER (15,6)	Null	-	
ALLOC_WATER_VOL_OUOM	VARCHAR2 (20)	Null	-	
CALC_ON_STREAM	NUMBER	Null	-	
CALC_ON_STREAM_OUOM	VARCHAR2 (20)	Null	-	
CASING_PRESS_13	NUMBER (15,6)	Null	-	
CASING_PRESS_13_OUOM	VARCHAR2 (20)	Null	-	
CASING_PRESS_9	NUMBER (15,6)	Null	-	
CASING_PRESS_9_OUOM	VARCHAR2 (20)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EMULSION	NUMBER (15,6)	Null	-	
EMULSION_OUOM	VARCHAR2 (20)	Null	-	
EXPIRY_DATE	DATE	Null	-	
GL_CHOKE	NUMBER	Null	-	
GL_CHOKE_OUOM	VARCHAR2 (20)	Null	-	
GL_MEASURED	NUMBER (15,6)	Null	-	
GL_MEASURED_OUOM	VARCHAR2 (20)	Null	-	
GL_PRESS	NUMBER (15,6)	Null	-	

GL_PRESS_OUOM	VARCHAR2 (20)	Null	-	
HP	NUMBER (15,6)	Null	-	
HP_OUOM	VARCHAR2 (20)	Null	-	
IP	NUMBER (15,6)	Null	-	
IP_OUOM	VARCHAR2 (20)	Null	-	
LAST_WELL_TEST	DATE	Null	-	
LP	NUMBER (15,6)	Null	-	
LP_OUOM	VARCHAR2	Null	-	
PPDM_GUID	VARCHAR2 (50)	Not null	-	
REMARK	VARCHAR2 (2000)	Null	-	
SAND_RATE	NUMBER (15,6)	Null	-	
SAND_RATE_OUOM	VARCHAR2 (20)	Null	-	
SEDIMENT	NUMBER (15,6)	Null	-	
SEDIMENT_OUOM	VARCHAR2 (20)	Null	-	
THEOR_GAS_VOL	NUMBER (15,6)	Null	-	
THEOR_GAS_VOL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_GL_VOL	NUMBER (15,6)	Null	-	
THEOR_GL_VOL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_OIL_VOL	NUMBER (15,6)	Null	-	
THEOR_OIL_VOL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_WATER_VOL	NUMBER (15,6)	Null	-	
THEOR_WATER_VOL_OUOM	VARCHAR2 (20)	Null	-	
VLP	NUMBER (15,6)	Null	-	
VLP_OUOM	VARCHAR2 (20)	Null	-	
WATER_CUT	NUMBER (15,6)	Null	-	
WATER_CUT_OUOM	VARCHAR2 (20)	Null	-	
WELL_HEAD_CHOKE	NUMBER (15,6)	Null	-	
WELL_HEAD_CHOKE_OUOM	VARCHAR2 (20)	Null	-	
WELL_HEAD_PRESSURE	NUMBER (15,6)	Null	-	
WELL_HEAD_PRESSURE_OUOM	VARCHAR2 (20)	Null	-	
WELL_HEAD_TEMPERATURE_F	NUMBER (15,6)	Null	-	
WELL_HEAD_TEMPERATURE_F_OUOM	VARCHAR2 (20)	Null	-	
WELL_TEST_CHOKE	NUMBER (15,6)	Null	-	
WELL_TEST_CHOKE_OUOM	VARCHAR2 (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW-CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	NUMBER	Null	-	
ALLOC_OIL_CORRECTED	NUMBER	Null	-	
AVG_SAND_RATE	VARCHAR2 (20)	Null	-	
AVG_SAND_RATE_OUOM	VARCHAR2 (20)	Null	-	
STRING_CODE	VARCHAR2 (30)	Null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
SHORT_NAME	VARCHAR2 (30)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	

OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

5. Tabel KLO_WELL_TEST

- Nama : KLO_WELL_TEST
- Deskripsi isi : Berisi data *well* tes khusus untuk KLO
- *Primary key* : *Uwi, string_id, test_date, test_num*
- *Foreign key* : *Uwi, string_id*

Tabel B. 5 Basis data well test

Nama Field	Type dan Length	Null	Default	Ket
UWI*/**	VARCHAR2 (20)	Not null	-	
STRING_ID*/**	VARCHAR2 (30)	Not null	-	
TEST_DATE*	DATE	Not null	-	
TEST_NUM*	VARCHAR2 (20)	Not null	-	
STRING_CODE	VARCHAR2 (30)	Null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
TEST_TIME	VARCHAR2 (20)	Not null	-	
TEST_TYPE	VARCHAR2 (20)	Not null	-	
TEST_DURATION	NUMBER	Null	-	
TEST_RESULT_CODE	VARCHAR2 (20)	Not null	-	
BASE_DEPTH	NUMBER (10,5)	Null	-	
TOP_DEPTH	NUMBER (10,5)	Null	-	
ZONE_NAME	VARCHAR2 (32)	Null	-	
SAND_NAME	VARCHAR2 (255)	Null	-	
PLATFORM	VARCHAR2 (32)	Null	-	
BSW_PERCENT	NUMBER	Null	-	
WH_CSG_PRESS	NUMBER	Null	-	
WH_PRESS	NUMBER	Null	-	
WH_TEMP	NUMBER	Null	-	
OIL_RATE	NUMBER	Null	-	
OIL_GRAVITY	NUMBER	Null	-	
WATER_CUT	NUMBER	Null	-	
WATER_RATE	NUMBER	Null	-	
GROSS_FLUID	NUMBER	Null	-	
SURF_CHOKE	NUMBER	Null	-	
GL_CHOKE	NUMBER	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

6. Tabel PROD_STRING

- Nama : PROD_STRING
- Deskripsi isi : Berisi *string* dari semua data *well*
- *Primary key* : *Uwi, string_id, string_code*

Tabel B. 6 Basis data prod string

Nama Field	Type dan Length	Null	Default	Ket
UWI*	VARCHAR2 (20)	Not null	-	
STRING_ID*	VARCHAR2 (30)	Not null	-	
STRING_CODE*	VARCHAR2 (30)	Not null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
SHORT_NAME	VARCHAR2 (30)	Null	-	
WELLHDR_CODE	VARCHAR2 (20)	Null	-	
STATUS_TYPE	VARCHAR2 (20)	Null	-	
CURRENT_STATUS	VARCHAR2 (20)	Null	-	
CURRENT_STATUS_DATE	DATE	Null	-	
STRING_TYPE	VARCHAR2 (20)	Null	-	
PROFILE_TYPE	VARCHAR2 (20)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
ON_PRODUCTION_DATE	DATE	Null	-	
PRODUCTION_METHOD	VARCHAR2 (20)	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

7. Tabel PPDM38_WELL_DAILY_DATA

- Nama : PPDM38_WELL_DAILY_DATA
- Deskripsi isi : Berisi data *well* harian
- *Primary key* : *Uwi, source, string_id*
- *Foreign key* : *Uwi, string_id*

Tabel B. 7 Basis data PPDM38 well daily data

<i>Nama Field</i>	<i>Type dan Length</i>	<i>Null</i>	<i>Default</i>	<i>Ket</i>
UWI**/	VARCHAR2 (20)	Not null	-	
SOURCE*	VARCHAR2 (20)	Not null	-	
STRING_ID**/	VARCHAR2 (20)	Not null	-	
STATUS_DATE	DATE	Null	-	
ACTIVE_IND	CHAR (1)	Null	-	
ALLOC_GAS_VOL	NUMBER (15,6)	Null	-	
ALLOC_GAS_VOL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_GL_VOL	NUMBER (15,6)	Null	-	
ALLOC_GL_VOL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_OIL_VOL	NUMBER (15,6)	Null	-	
ALLOC_OIL_VOL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_WATER_VOL	NUMBER (15,6)	Null	-	
ALLOC_WATER_VOL_OUOM	VARCHAR2 (20)	Null	-	
CALC_ON_STREAM	NUMBER	Null	-	
CALC_ON_STREAM_OUOM	VARCHAR2 (20)	Null	-	
CASING_PRESS_13	NUMBER (15,6)	Null	-	
CASING_PRESS_13_OUOM	VARCHAR2 (20)	Null	-	
CASING_PRESS_9	NUMBER (15,6)	Null	-	
CASING_PRESS_9_OUOM	VARCHAR2 (20)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EMULSION	NUMBER (15,6)	Null	-	
EMULSION_OUOM	VARCHAR2 (20)	Null	-	
EXPIRY_DATE	DATE	Null	-	
GL_CHOKE	NUMBER (15,6)	Null	-	
GL_CHOKE_OUOM	VARCHAR2 (20)	Null	-	
GL_MEASURED	NUMBER (15,6)	Null	-	
GL_MEASURED_OUOM	VARCHAR2 (20)	Null	-	
GL_PRESS	NUMBER (15,6)	Null	-	
GL_PRESS_OUOM	VARCHAR2 (20)	Null	-	
HP	NUMBER (15,6)	Null	-	
HP_OUOM	VARCHAR2 (20)	Null	-	
IP	NUMBER (15,6)	Null	-	
IP_OUOM	VARCHAR2 (20)	Null	-	
LAST_WELL_TEST	DATE	Null	-	
LP	NUMBER (15,6)	Null	-	
LP_OUOM	VARCHAR2 (20)	Null	-	
PPDM_GUID	VARCHAR2 (50)	Not null	-	
REMARK	VARCHAR2 (2000)	Null	-	
SAND_RATE	NUMBER (15,6)	Null	-	

SAND_RATE_OUOM	VARCHAR2 (20)	Null	-	
SEDIMENT	NUMBER (15,6)	Null	-	
SEDIMENT_OUOM	VARCHAR2 (20)	Null	-	
THEOR_GAS_VOL	NUMBER (15,6)	Null	-	
THEOR_GAS_VOL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_GL_VOL	NUMBER (15,6)	Null	-	
THEOR_GL_VOL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_OIL_VOL	NUMBER (15,6)	Null	-	
THEOR_OIL_VOL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_WATER_VOL	NUMBER (20)	Null	-	
THEOR_WATER_VOL_OUOM	VARCHAR2 (20)	Null	-	
VLP	NUMBER (15,6)	Null	-	
VLP_OUOM	VARCHAR2 (20)	Null	-	
WATER_CUT	NUMBER (15,6)	Null	-	
WATER_CUT_OUOM	VARCHAR2 (20)	Null	-	
WELL_HEAD_CHOKE	NUMBER (15,6)	Null	-	
WELL_HEAD_CHOKE_OUOM	VARCHAR2 (20)	Null	-	
WELL_HEAD_PRESSURE	NUMBER (15,6)	Null	-	
WELL_HEAD_PRESSURE_OUOM	VARCHAR2 (20)	Null	-	
WELL_HEAD_TEMPERATURE_F	NUMBER (15,6)	Null	-	
WELL_HEAD_TEMPERATURE_F	VARCHAR2 (20)	Null	-	
WELL_TEST_CHOKE	NUMBER (6)	Null	-	
WELL_TEST_CHOKE_OUOM	VARCHAR2 (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	NUMBER	Null	-	
ALLOC_OIL_CORRECTED	NUMBER	Null	-	
AVG_SAND_RATE	NUMBER	Null	-	
AVG_SAND_RATE_OUOM	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

8. Tabel PPDM38_PDEN_FLOW_MEASUREMENT

- Nama : PPDM38_PDEN_FLOW_MEASUREMENT
- Deskripsi isi : Berisi pengukuran data *well*
- Primary key : *Pden_id, pden_type, pden_source, product_type, measurement_date, amendment_seq_no*

Tabel B. 8 Basis data PPDM38 pden flow measurement

Nama Field	Type dan Length	Null	Default	Ket
PDEN_ID*	VARCHAR2 (40)	Not null	-	
PDEN_TYPE*	VARCHAR2 (30)	Not null	-	
PDEN_SOURCE*	VARCHAR2 (20)	Not null	-	
PRODUCT_TYPE *	VARCHAR2 (20)	Not null	-	
MEASUREMENT_DATE*	DATE	Not null	-	
AMENDMENT_SEQ_NO*	NUMBER (8)	Not null	-	
ACTIVE_IND	VARCHAR2 (1)	Null	-	
ALLOC_FIXED	VARCHAR2 (31)	Null	-	
ALLOC_NET_MASS	NUMBER (16,4)	Null	-	
ALLOC_NET_MASS_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_NET_VOL	NUMBER	Null	-	
ALLOC_NET_VOL_MTD	NUMBER	Null	-	
ALLOC_NET_VOL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_NET_VOL_YTD	NUMBER	Null	-	
ALLOC_PERIOD	VARCHAR2 (32)	Null	-	
AMEND_REASON	VARCHAR2 (20)	Null	-	
BS_W	NUMBER	Null	-	
BS_W_OUOM	VARCHAR2 (20)	Null	-	
CASING_PRESSURE	NUMBER (8,2)	Null	-	
CASING_PRESSURE_OUOM	VARCHAR2 (20)	Null	-	
CHOKE_POSITION	NUMBER (10,5)	Null	-	
CHOKE_SIZE	NUMBER (10,5)	Null	-	
CHOKE_SIZE_OUOM	VARCHAR2 (20)	Null	-	
CLOSING_READING	NUMBER (14,4)	Null	-	
CLOSING_READING_OUOM	VARCHAR2 (20)	Null	-	
DATE_FORMAT_DESC	VARCHAR2 (20)	Null	-	
DATE_METER_FACTOR	DATE	Null	-	
DEW_POINT_F	NUMBER (8,2)	Null	-	
DEW_POINT_F_OUOM	VARCHAR2 (20)	Null	-	
DIFF_PRESSURE_OUOM	VARCHAR2 (20)	Null	-	
DIFFERENTIAL_PRESSURE	NUMBER (8,2)	Null	-	
DISCHARGE_TO_OVERBOARD	NUMBER (14,4)	Null	-	
DISCHARGE_TO_OVERBOARD_OUOM	VARCHAR2 (20)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EMUL_SED_CONTENT	NUMBER 98,2)	Null	-	
EMUL_SED_CONTENT_OUOM	VARCHAR2 (20)	Null	-	
EXPIRY_DATE	DATE	Null	-	

FLOW_RATE	NUMBER	Null	-	
FLOW_RATE_MTD	NUMBER	Null	-	
FLOW_RATE_OUOM	VARCHAR2 (20)	Null	-	
FLOW_RATE_YTD	NUMBER	Null	-	
MEAS_TEMPERATURE	NUMBER (7,2)	Null	-	
MEAS_TEMPERATURE_OUOM	VARCHAR2 (20)	Null	-	
MEASUREMENT_TYPE	VARCHAR2 (20)	Null	-	
METER_FACTOR	VARCHAR2 (20)	Null	-	
OIL_GRAVITY	NUMBER (8,2)	Null	-	
OIL_GRAVITY_OUOM	VARCHAR2 (20)	Null	-	
OIL_VOL_CONTENT_AFTER	NUMBER (14,4)	Null	-	
OIL_VOL_CONTENT_AFTER_OUOM	VARCHAR2 (20)	Null	-	
OIL_VOL_CONTENT_BEFORE	NUMBER (14,4)	Null	-	
OIL_VOL_CONTENT_BEFORE_OUOM	VARCHAR2 (20)	Null	-	
ON_STRM_HRS	NUMBER (14,2)	Null	-	
ON_STRM_HRS_OUOM	VARCHAR2 (20)	Null	-	
OPENING_READING	NUMBER (14,4)	Null	-	
OPENING_READING_OUOM	VARCHAR2 (20)	Null	-	
OVERRIDE_READING	NUMBER (14,4)	Null	-	
OVERRIDE_READING_OUOM	VARCHAR2 (20)	Null	-	
POSTED_DATE	DATE	Null	-	
PPDM_GUID	VARCHAR2 (50)	Not null	-	
PRODUCTION_TIME	NUMBER (14,4)	Null	-	
PRODUCTION_TIME_OUOM	VARCHAR2 (20)	Null	-	
PRODUCTION_VOLUME	NUMBER (14,4)	Null	-	
PRODUCTION_VOLUME_OUOM	VARCHAR2 (20)	Null	-	
PRODUCTION_VOLUMN_UOM	VARCHAR2 (20)	Null	-	
REMARK	VARCHAR2 (2000)	Null	-	
SOURCE	VARCHAR2 (20)	Null	-	
STATIC_PRESSURE	NUMBER (8,2)	Null	-	
STATIC_PRESSURE_OUOM	VARCHAR2 (20)	Null	-	
THEOR_NET_MASS	NUMBER (16,4)	Null	-	
THEOR_NET_MASS_OUOM	VARCHAR2 (20)	Null	-	
THEOR_NET_VOL	NUMBER	Null	-	
THEOR_NET_VOL_MTD	NUMBER	Null	-	
THEOR_NET_VOL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_NET_VOL_YTD	NUMBER	Null	-	
TUBING_PRESSURE	NUMBER (8,2)	Null	-	
TUBING_PRESSURE_OUOM	VARCHAR2 (20)	Null	-	
WATER_PH	NUMBER (14,4)	Null	-	
WATER_PH_OUOM	VARCHAR2 (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	

ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

9. Tabel PPDM38_WELL_DEFERMENT_DAILY

- Nama : PPDM38_WELL_DEFERMENT_DAILY
- Deskripsi isi : Berisi data well harian yang mengalami penundaan
- Primary key : Pden_id, pden_type
- Foreign key : Pden_id, pden_type

Tabel B. 9 Basis data PPDM38 well deferment daily

Nama Field	Type dan Length	Null	Default	Ket
PDEN_ID*/**	VARCHAR2 (40)	Not null	-	
PDEN_TYPE*/**	VARCHAR2 (30)	Not null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
WELL_NAME	VARCHAR2 (255)	Null	-	
UWI	VARCHAR2 (20)	Not null	-	
STRING_ID	VARCHAR2 (20)	Not null	-	
FIELD	VARCHAR2 (32)	Null	-	
FACILITY_NAME	VARCHAR2 (32)	Null		
PERIOD_TYPE	VARCHAR2 (20)	Null	-	
VOLUME_DATE	DATE	Not null	-	
DEFERED_CON_PROD	NUMBER (16,4)	Null	-	
DEFERED_CON_PROD_OUOM	VARCHAR2 (20)	Null	-	
DEFERED_FLUID_PROD	NUMBER (16,4)	Null	-	
DEFERED_FLUID_PROD_OUOM	VARCHAR2 (20)	Null	-	
DEFERED_GAS_PROD	NUMBER (16,4)	Null	-	
DEFERED_GAS_PROD_OUOM	VARCHAR2 (20)	Null	-	
DEFERED_OIL_PROD	NUMBER (16,4)	Null	-	
DEFERED_OIL_PROD_OUOM	VARCHAR2 (20)	Null		
DEFERED_OIL_PROD_CORRECTED	NUMBER	Null	-	
DEFERED_WTR_PROD	NUMBER (16,4)	Null	-	
DEFERED_WTR_PROD_OUOM	VARCHAR2 (20)	Null	-	
DURATION_HOUR_PER_DAY	NUMBER (16,4)	Null	-	
DURATION_HOUR_SINCE_OFF	NUMBER (16,4)	Null	-	
DURATION_DAY_SINCE_OFF	NUMBER (16,4)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
SURFACE_OR_SUBSURFACE	VARCHAR2 (20)	Null	-	
PROACTIVE_OR_REACTIVE	VARCHAR2 (20)	Null	-	
EVENT_SYSTEM	VARCHAR2 (16)	Null	-	
EVENT_CATEGORY	VARCHAR2 (16)	Null	-	

EVENT_REASON	VARCHAR2 (16)	Null	-	
KLO_CODE	VARCHAR2 (16)	Null		
REMARK	VARCHAR2 (2000)	Null	-	
START_DATE_EVENT	DATE	Null	-	
END_DATE_EVENT	DATE	Null	-	
POTENTIAL_METHOD	VARCHAR2 (32)	Null	-	
WORK_ORDER_ID	VARCHAR2 (32)	Null	-	

10. Tabel PPDM38_PDEN_VOL_PROD_DAY

- Nama : PPDM38_PDEN_VOL_PROD_DAY
- Deskripsi isi : Berisi produksi *volume well*
- Primary key : *Pden_id, pden_type, pden_source, volume method, activity_type, period_type, colume_date, amendment_seq_no, string_code*
- Foreign key : *Pden_id, pden_type, pden_source, amendment_seq_no, string_code*

Tabel B. 10 Basis data PPDM38 pden vol prod day

Nama Field	Type dan Length	Null	Default	Ket
PDEN_ID*/**	VARCHAR2 (40)	Not null	-	
PDEN_TYPE*/**	VARCHAR2 (30)	Not null	-	
PDEN_SOURCE*/**	VARCHAR2 (20)	Not null	-	
VOLUME_METHOD*	VARCHAR2 (20)	Not null	-	
ACTIVITY_TYPE*	VARCHAR2 (20)	Not null	-	
PERIOD_TYPE*	VARCHAR2 (20)	Not null	-	
VOLUME_DATE*	DATE	Not null	-	
AMENDMENT_SEQ_NO*/**	NUMBER (8)	Not null	-	
STRING_CODE*/**	VARCHAR2 (30)	Not null	-	
ACTIVE_IND	VARCHAR2 (1)	Null	-	
ALLOCATION_FACTOR	NUMBER (10,5)	Null	-	
ALLOC_CON	NUMBER (16,4)	Null	-	
ALLOC_CON_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_FLUID	NUMBER (16,4)	Null	-	
ALLOC_FLUID_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_OIL	NUMBER (16,4)	Null	-	
ALLOC_OIL_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_GAS	NUMBER (16,4)	Null	-	
ALLOC_GAS_OUOM	VARCHAR2 (20)	Null	-	
ALLOC_WTR	NUMBER (16,4)	Null	-	
ALLOC_WTR_OUOM	VARCHAR2 (20)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
PERIOD_ON_PRODUCTION	NUMBER (10,5)	Null	-	
PERIOD_ON_PRODUCTION_OUOM	VARCHAR2 (20)	Null	-	

PPDM_GUID	VARCHAR2 (50)	Not null	-	
REMARK	VARCHAR2 (2000)	Null	-	
SOURCE	VARCHAR2 (20)	Null	-	
THEOR_CON	NUMBER (16,4)	Null	-	
THEOR_CON_OUOM	VARCHAR2 (20)	Null	-	
THEOR_FLUID	NUMBER (16,4)	Null	-	
THEOR_FLUID_OUOM	VARCHAR2 (20)	Null	-	
THEOR_OIL	NUMBER (16,4)	Null	-	
THEOR_OIL_OUOM	VARCHAR2 (20)	Null	-	
THEOR_GAS	NUMBER (16,4)	Null	-	
THEOR_GAS_OUOM	VARCHAR2 (20)	Null	-	
THEOR_WTR	NUMBER (16,4)	Null	-	
THEOR_WTR_OUOM	VARCHAR2 (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	
ALLOC_GAS_LIFT	NUMBER	Null	-	
ALLOC_GAS_LIFT_OUOM	VARCHAR2 (20)	Null	-	
THEOR_GAS_LIFT	NUMBER	Null	-	
THEOR_GAS_LIFT_OUOM	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

11. Tabel PPDM38_PROD_STRING_ALIAS

- Nama : PPDM38_PROD_STRING_ALIAS
- Deskripsi isi : Berisi *string* dari semua data *well*
- Primary key : *Uwi, prod_string_source, string_id, source, alias_id*
- Foreign key : *Uwi, string_id*

Tabel B. 11 Basis data PPDM38 prod string alias

Nama Field	Type dan Length	Null	Default	Ket
UWI*/**	VARCHAR2 (20)	Not null	-	
PROD_STRING_SOURCE*	VARCHAR2 (20)	Not null	-	
STRING_ID*/**	VARCHAR2 (20)	Not null	-	
SOURCE*	VARCHAR2 (20)	Not null	-	
ALIAS_ID*	VARCHAR2 (40)	Not null	-	
ACTIVE_IND	VARCHAR2 (1)	Null	-	
ALIAS_CODE	VARCHAR2 (32)	Null	-	
ALIAS_FULL_NAME	VARCHAR2 (255)	Null	-	
ALIAS_REASON_TYPE	VARCHAR2 (20)	Null	-	
ALIAS_SHORT_NAME	VARCHAR2 (30)	Null	-	
ALIAS_TYPE	VARCHAR2 (20)	Null	-	
AMENDED_DATE	DATE	Null	-	
CREATED_DATE	DATE	Null	-	
EFFECTIVE_DATE	DATE	Null	-	

EXPIRY_DATE	DATE	Null	-	
LANGUAGE_ID	VARCHAR2 (20)	Null	-	
LOCATION_TYPE	VARCHAR2 (20)	Null	-	
ORIGINAL_IND	VARCHAR2 (1)	Null	-	
OWNER_BA_ID	VARCHAR2 (20)	Null	-	
PPDM_GUID	VARCHAR2 (38)	Not null	-	
PREFERRED_IND	VARCHAR2 (1)	Null	-	
REASON_DESC	VARCHAR2 (2000)	Null	-	
REMARK	VARCHAR2 (2000)	Null	-	
SOURCE_DOCUMENT	VARCHAR2 (20)	Null	-	
STRUCKOFF_DATE	DATE	Null	-	
SW_APPLICATION_ID	VARCHAR2 (20)	Null	-	
USE_RULE	VARCHAR2 (2000)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

12. Tabel PPDM38_PDEN_VOL_PROD_DEFER_DAY

- Nama : PPDM38_PDEN_VOL_PROD_DEFER_DAY
- Deskripsi isi : Berisi data *volume* produksi harian yang mengalami penundaan
- *Primary key* : Pden_id, pden_type, pden_source, volume_method, activity_type, period_type, volume_date, amendment_seq_no, start_date_event, string_code
- *Foreign key* : Pden_id, pden_type, pden_source, amendment_seq_no, string_code

Tabel B. 12 Basis data PPDM38 pden vol prod defer day

Nama Field	Type dan Length	Null	Default	Ket
PDEN_ID*/**	VARCHAR2 (40)	Not null	-	
PDEN_TYPE*/**	VARCHAR2 (30)	Not null	-	
PDEN_SOURCE*/**	VARCHAR2 (20)	Not null	-	
VOLUME_METHOD*	VARCHAR2 (20)	Not null	-	
ACTIVITY_TYPE*	VARCHAR2 (20)	Not null	-	
PERIOD_TYPE*	VARCHAR2 (20)	Not null	-	
VOLUME_DATE*	DATE	Not null	-	
AMENDMENT_SEQ_NO*/**	NUMBER (8)	Not null	-	
START_DATE_EVENT*	DATE	Not null	-	
STRING_CODE*/**	VARCHAR2 (30)	Not null		
END_DATE_EVENT	DATE	Null	-	
ACTIVE_IND	VARCHAR2 (1)	Null	-	
DEFERED_CON_PROD	NUMBER (16,4)	Null	-	

DEFERED_CON_PROD_OUOM	VARCHAR2 (20)	Null	-	
DEFERED_FLUID_PROD	NUMBER (16,4)	Null	-	
DEFERED_FLUID_PROD_OUOM	VARCHAR2 (20)	Null	-	
DEFERED_GAS_PROD	NUMBER (16,4)	Null	-	
DEFERED_GAS_PROD_OUOM	VARCHAR2 (20)	Null	-	
DEFERED_OIL_PROD	NUMBER (16,4)	Null	-	
DEFERED_OIL_PROD_OUOM	VARCHAR2 (20)	Null	-	
DEFERED_WTR_PROD	NUMBER (16,4)	Null	-	
DEFERED_WTR_PROD_OUOM	VARCHAR2 (20)	Null	-	
DURATION_HOUR_PER_DAY	NUMBER (16,4)	Null	-	
DURATION_HOUR_SINCE_OFF	NUMBER (16,4)	Null	-	
DURATION_DAY_SINCE_OFF	NUMBER (16,4)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
EVENT_CATEGORY	VARCHAR2 (16)	Null	-	
EVENT_REASON	VARCHAR2 (16)	Null	-	
EVENT_SYSTEM	VARCHAR2 (32)	Null	-	
POTENTIAL_METHOD	VARCHAR2 (32)	Null	-	
PROACTIVE_OR_REACTIVE	VARCHAR2 (20)	Null	-	
SURFACE_OR_SUBSURFACE	VARCHAR2 (20)	Null	-	
PPDM_GUID	VARCHAR2 (50)	Not null	-	
REMARK	VARCHAR2 (2000)	Null	-	
SOURCE	VARCHAR2 (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	
WORK_ORDER_ID	VARCHAR2 (32)	Null	-	
EVENT_CATEGORY_NAME	VARCHAR2 (255)	Null	-	
EVENT_REASON_NAME	VARCHAR2 (255)	Null	-	
EVENT_SYSTEM_NAME	VARCHAR2 (255)	Null	-	
DEFERED_OIL_PROD_CORRECTED	NUMBER	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

13. Tabel TIME_DIM

- Nama : TIME_DIM
- Deskripsi isi : Berisi waktu dan tanggal
- *Primary key* : -

Tabel B. 13 Basis data time dim

<i>Nama Field</i>	<i>Type dan Length</i>	<i>Null</i>	<i>Default</i>	<i>Ket</i>
DAYTIME	DATE	Not null	-	
DAY_OF_WEEK	NUMBER (1)	Not null	-	
DAY_NAME	VARCHAR2 (27)	Not null	-	
DAY_OF_MONTH	NUMBER	Not null	-	
MONTH_NUMBER	NUMBER (2)	Not null	-	
MONTH_NAME	VARCHAR2 (27)	Not null	-	
END_OF_MONTH	NUMBER (1)	Not null	-	
QUARTER_NUMBER	NUMBER (1)	Not null	-	
QUARTER_NAME	VARCHAR2 (6)	Not null	-	
DAY_OF_YEAR	NUMBER (3)	Not null	-	
YEAR	NUMBER (4)	Not null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

14. Tabel STRING_STATUS_HIST

- Nama : STRING_STATUS_HIST
- Deskripsi isi : Berisi *string* status *historical*
- *Primary key* : *Uwi, string_id, start_time, string_code*
- *Foreign key* : *Uwi, string_id, string_code*

Tabel B. 14 Basis data string status hist

<i>Nama Field</i>	<i>Type dan Length</i>	<i>Null</i>	<i>Default</i>	<i>Ket</i>
UWI*/**	VARCHAR2 (20)	Not null	-	
STRING_ID*/**	VARCHAR2 (30)	Not null	-	
START_TIME*	DATE	Not null	-	
END_TIME	DATE	Null	-	
STRING_CODE */**	VARCHAR2 (30)	Null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
ALT_NAME	VARCHAR2 (30)	Null	-	
GRID_NAME	VARCHAR2 (30)	Null	-	
STATUS	VARCHAR2 (20)	Null	-	
STATUS_TYPE	VARCHAR2 (20)	Null	-	
PROD_STRING_TYPE	VARCHAR2 (20)	Null	-	
PRODUCTION_METHOD	VARCHAR2 (20)	Null	-	
EVENT_CATEGORY	VARCHAR2 (16)	Null	-	
EVENT_REASON	VARCHAR2 (16)	Null	-	
EVENT_SYSTEM	VARCHAR2 (16)	Null	-	
PROACTIVE_OR_REACTIVE	VARCHAR2 (20)	Null	-	
SURFACE_OR_SUBSURFACE	VARCHAR2 (20)	Null	-	

REMARK	VARCHAR2 (500)	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	
ASSET_AREA	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
AREA	VARCHAR2 (32)	Null	-	
ARSE	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

15. Tabel PPDM38_PROD_STR_STAT_HIST

- Nama : PPDM38_PROD_STR_STAT_HIST
- Deskripsi isi : Berisi data produksi *string status historical*
- Primary key : Uwi, prod_stirng_source, string_id, status_date
- Foreign key : Uwi, string_id

Tabel B. 15 Basis data PPDM38 prod str stat hist

Nama Field	Type dan Length	Null	Default	Ket
UWI*/**	VARCHAR2 (20)	Not null	-	
PROD_STRING_SOURCE*	VARCHAR2 (20)	Not null	-	
STRING_ID*/**	VARCHAR2 (20)	Not null	-	
STATUS_DATE*	DATE	Not null	-	
ACTIVE_IND	VARCHAR2 (1)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
END_TIME	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
PERCENT_CAPABILITY	NUMBER (15,12)	Null	-	
PPDM_GUID	VARCHAR2 (50)	Not null	-	
REMARK	VARCHAR2 (2000)	Null	-	
SOURCE	VARCHAR2 (20)	Null	-	
START_TIME	DATE	Null	-	
STATUS	VARCHAR2 (20)	Null	-	
STATUS_TYPE	VARCHAR2 (20)	Null	-	
TIMEZONE	VARCHAR2 (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	

ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	
PROD_STRING_TYPE	VARCHAR2 (20)	Null	-	
SURFACE_OR_SUBSURFACE	VARCHAR2 (20)	Null	-	
PROACTIVE_OR_REACTIVE	VARCHAR2 (20)	Null	-	
EVENT_CATEGORY	VARCHAR2 (16)	Null	-	
EVENT_REASON	VARCHAR2 (16)	Null	-	
EVENT_SYSTEM	VARCHAR2 (16)	Null	-	
PRODUCTION_METHOD	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

16. Tabel WELL_JOB

- Nama : WELL_JOB
- Deskripsi isi : Berisi pekerjaan yang ada di *well*
- *Primary key* : *Work_order_id, uwi, string_id*
- *Foreign key* : *Uwi, string_id*

Tabel B. 16 Basis data well job

Nama Field	Type dan Length	Null	Default	Ket
WORK_ORDER_ID*	VARCHAR2 (32)	Not null	-	
UWI*/**	VARCHAR2 (20)	Not null	-	
STRING_ID*/**	VARCHAR2 (30)	Not null	-	
STRING_CODE	VARCHAR2 (30)	Null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
ALT_NAME	VARCHAR2 (30)	Null	-	
PPDM_STRING_CODE	VARCHAR2 (30)	Null	-	
PROD_STRING_TYPE	VARCHAR2 (20)	Null	-	
JOB_NUMBER	VARCHAR2 (50)	Null	-	
JOB_STATUS	VARCHAR2 (50)	Null	-	
COST_CENTER	VARCHAR2 (50)	Null	-	
SUBMITTED_BY	VARCHAR2 (50)	Null	-	
REQUEST_DATE	DATE	Null	-	
START_DATE	DATE	Null	-	
END_DATE	DATE	Null	-	
JOB_CATEGORY	VARCHAR2 (50)	Null	-	
JOB_TYPE	VARCHAR2 (50)	Null	-	
JOB_DETAIL_CODE	VARCHAR2 (5)	Null	-	
JOB_DETAIL_DESCRIPTION	VARCHAR2 (50)	Null	-	
ACTUAL_COST	NUMBER	Null	-	
ACTUAL_DURATION	NUMBER	Null	-	
ESTIMATE_COST	NUMBER	Null	-	
ESTIMATE_DURATION	NUMBER	Null	-	
ESTIMATE_RIG_DAY	NUMBER	Null	-	
TARGET_OIL	NUMBER	Null	-	
TARGET_GAS	NUMBER	Null	-	
TARGET_WATER	NUMBER	Null	-	

TARGET_CONDENSATE	NUMBER	Null	-	
WELL_OFF_DATE	DATE	Null	-	
WELL_ON_DATE	DATE	Null	-	
REMARK	VARCHAR2 (2000)	Null	-	
RIG_CODE	VARCHAR2 (40)	Null	-	
RIG_NAME	VARCHAR2 (255)	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	Null	-	
LICENCE	VARCHAR2 (32)	Null	-	
OP_AREA_CODE	VARCHAR2 (32)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	Null	-	
ASSET_AREA	VARCHAR2 (32)	Null	-	
FIELD	VARCHAR2 (32)	Null	-	
AREA	VARCHAR2 (32)	Null	-	
ARSE	VARCHAR2 (32)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

17. Tabel PPDM38_WORK_ORDER

- Nama : PPDM38_WORK_ORDER
- Deskripsi isi : Berisi urutan pekerjaan *well*
- *Primary key* : *Work_order_id*

Tabel B. 17 Basis data PPDM38 work order

Nama Field	Type dan Length	Null	Default	Ket
WORK_ORDER_ID*	VARCHAR2 (32)	Not null	-	
ACTIVE_IND	VARCHAR2 (1)	Null	-	
COMPLETE_DATE	DATE	Null	-	
DUE_DATE	DATE	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
FINAL_BILLING_DATE	DATE	Null	-	
INSTRUCTIONS	VARCHAR2 (2000)	Null	-	
PPDM_GUID	VARCHAR2 (38)	Not null	-	
REMARK	VARCHAR2 (2000)	Null	-	
REQUEST_DATE	DATE	Null	-	
RUSH_IND	VARCHAR2 (1)	Null	-	
SOURCE	VARCHAR (20)	Null	-	

WORK_ORDER_NUMBER	VARCHAR2 (60)	Null	-	
WORK_ORDER_TYPE	VARCHAR (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR (20)	Null	-	
JOB_CATEGORY	VARCHAR2 (50)	Null	-	
JOB_TYPE	VARCHAR2 (50)	Null	-	
JOB_DETAIL	VARCHAR2 (50)	Null	-	
JOB_STATUS	VARCHAR2 (50)	Null	-	
ACTUAL_COST	NUMBER	Null	-	
ACTUAL_DURATION	NUMBER	Null	-	
ESTIMATE_COST	NUMBER	Null	-	
ESTIMATE_DURATION	NUMBER	Null	-	
ESTIMATE_RIG_DAY	NUMBER	Null	-	
TARGET_OIL	NUMBER	Null	-	
TARGET_GAS	NUMBER	Null	-	
TARGET_WATER	NUMBER	Null	-	
TARGET_CONDENSATE	NUMBER	Null	-	
COST_CENTER	VARCHAR2 (50)	Null	-	
SUBMITTED_BY	VARCHAR2 (50)	Null	-	
WELL_OFF_DATE	DATE	Null	-	
WELL_ON_DATE	DATE	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

18. Tabel PPDM38_WORK_ORDER_COMPONENT

- Nama : PPDM38_WORK_ORDER_COMPONENT
- Deskripsi isi : Berisi komponen urutan pekerjaan *well*
- *Primary key* : *Work_order_id, component_id, uwi*
- *Foreign key* : *Work_order_id, uwi*

Tabel B. 18 Basis data PPDM38 work order component

Nama Field	Type dan Length	Null	Default	Ket
WORK_ORDER_ID*/**	VARCHAR2 (32)	Not null	-	
COMPONENT_ID*	VARCHAR2 (20)	Not null	-	
ACTIVE_IND	VARCHAR2 (1)	Null	-	
ACTIVITY_OBS_NO	NUMBER (8)	Null	-	
ANALYSIS_ID	VARCHAR2 (20)	Null	-	
APPLICATION_ID	VARCHAR2 (20)	Null	-	
AREA_ID	VARCHAR2 (20)	Null	-	
AREA_TYPE	VARCHAR2 (20)	Null	-	
AUTHORITY_ID	VARCHAR2 (20)	Null	-	
BA_ORGANIZATION_ID	VARCHAR2 (20)	Null	-	
BA_ORGANIZATION_SEQ_NO	NUMBER (8)	Null	-	
BUSINESS_ASSOCIATE	VARCHAR2 (20)	Null	-	
CATALOGUE_ADDITIVE_ID	VARCHAR2 (20)	Null	-	
CATALOGUE_EQUIP_ID	VARCHAR2 (20)	Null	-	

CIRC_ID	VARCHAR2 (20)	Null	-	
CLASSIFICATION_SYSTEM_ID	VARCHAR2 (20)	Null	-	
CLASS_LEVEL_ID	VARCHAR2 (20)	Null	-	
COMPONENT_TYPE	VARCHAR2 (20)	Null	-	
CONSENT_ID	VARCHAR2 (20)	Null	-	
CONSULT_ID	VARCHAR2 (20)	Null	-	
CONTEST_ID	VARCHAR2 (20)	Null	-	
CONTRACT_ID	VARCHAR2 (20)	Null	-	
ECOZONE_ID	VARCHAR2 (20)	Null	-	
EFFECTIVE_DATE	VARCHAR2 (20)	Null	-	
EMPLOYEE_BA_ID	VARCHAR2 (20)	Null	-	
EMPLOYEE_OBS_NO	NUMBER (8)	Null	-	
EMPLOYER_BA_ID	VARCHAR2 (20)	Null	-	
ENTITLEMENT_ID	VARCHAR2 (20)	Null	-	
EQUIPMENT_ID	VARCHAR2 (20)	Null	-	
EQUIP_MAINT_ID	VARCHAR2 (20)	Null	-	
EXPIRY_DATE	DATE	Null	-	
FACILITY_ID	VARCHAR2 (32)	Null	-	
FACILITY_MAINTAIN_ID	VARCHAR2 (20)	Null	-	
FACILITY_TYPE	VARCHAR2 (20)	Null	-	
FIELD_ID	VARCHAR2 (20)	Null	-	
FINANCE_ID	VARCHAR2 (20)	Null	-	
FOSSIL_ID	VARCHAR2 (20)	Null	-	
INCIDENT_ID	VARCHAR2 (20)	Null	-	
INCIDENT_SET_ID	VARCHAR2 (20)	Null	-	
INCIDENT_TYPE	VARCHAR2 (20)	Null	-	
INFORMATION_ITEM_ID	VARCHAR2 (20)	Null	-	
INFO_ITEM_TYPE	VARCHAR2 (20)	Null	-	
INSPECTION_ID	VARCHAR2 (20)	Null	-	
INSTRUMENT_ID	VARCHAR2 (20)	Null	-	
INTEREST_SET_ID	VARCHAR2 (20)	Null	-	
INTEREST_SET_SEQ_NO	NUMBER (8)	Null	-	
JURISDICTION	VARCHAR2 (20)	Null	-	
LAND_RIGHT_ID	VARCHAR2 (20)	Null	-	
LAND_RIGHT_TYPE	VARCHAR2 (20)	Null	-	
LAND_SALE_NUMBER	VARCHAR2 (20)	Null	-	
LANGUAGE	VARCHAR2 (20)	Null	-	
LITHOLOGY_LOG_ID	VARCHAR2 (20)	Null	-	
LITH_LOG_SOURCE	VARCHAR2 (20)	Null	-	
NOTIFICATION_ID	VARCHAR2 (20)	Null	-	
OBLIGATION_ID	VARCHAR2 (20)	Null	-	
OBLIGATION_SEQ_NO	NUMBER (8)	Null	-	
ORIGIN_SEQ_NO	NUMBER (8)	Null	-	
PALEO_SUMMARY_ID	VARCHAR2 (20)	Null	-	
PDEN_ID	VARCHAR2 (40)	Null	-	
PDEN_SOURCE	VARCHAR2 (20)	Null	-	
PDEN_TYPE	VARCHAR2 (20)	Null	-	
PHYSICAL_ITEM_ID	VARCHAR2 (20)	Null	-	
POOL_ID	VARCHAR2 (20)	Null	-	
PPDM_GUID	VARCHAR2 (38)	Not null	-	

PPDM_SYSTEM_ID	VARCHAR2 (20)	Null	-	
PPDM_TABLE_NAME	VARCHAR2 (30)	Null	-	
PRODUCT_TYPE	VARCHAR2 (20)	Null	-	
PROD_STRING_ID	VARCHAR2 (20)	Null	-	
PROD_STRING_SOURCE	VARCHAR2 (20)	Null	-	
PROJECT_ID	VARCHAR2 (20)	Null	-	
RATE_SCHEDULE_ID	VARCHAR2 (20)	Null	-	
REFERENCED_GUID	VARCHAR2 (38)	Null	-	
REFERENCED_SYSTEM_ID	VARCHAR2 (20)	Null	-	
REFERENCED_TABLE_NAME	VARCHAR2 (30)	Null	-	
REMARK	VARCHAR2 (2000)	Null	-	
RESENT_ID	VARCHAR2 (20)	Null	-	
RESERVE_CLASS_ID	VARCHAR2 (20)	Null	-	
SAMPLE_ANAL_SOURCE	VARCHAR2 (20)	Null	-	
SEIS_SET_ID	VARCHAR2 (20)	Null	-	
SEIS_SET_TYPE	VARCHAR2 (30)	Null	-	
SEIS_TRANSACTION_ID	VARCHAR2 (20)	Null	-	
SF_TYPE	VARCHAR2 (30)	Null	-	
SOURCE	VARCHAR2 (20)	Null	-	
SPATIAL_DESCRIPTION_ID	VARCHAR2 (20)	Null	-	
SPATIAL_OBS_NO	NUMBER	Null	-	
STORE_ID	VARCHAR2 (20)	Null	-	
STORE_STRUCTURE_OBS_NO	NUMBER	Null	-	
STRAT_NAME_SET_ID	VARCHAR2 (20)	Null	-	
STRAT_UNIT_ID	VARCHAR2 (20)	Null	-	
SUPPORT_FACILITY_ID	VARCHAR2 (20)	Null	-	
THESAURUS_ID	VARCHAR2 (20)	Null	-	
THESAURUS_WORD_ID	VARCHAR2 (20)	Null	-	
TRANSACTION_TYPE	VARCHAR2 (20)	Null	-	
UWI * / **	VARCHAR2 (20)	Not null	-	
WELL_ACTIVITY_SET_ID	VARCHAR2 (20)	Null	-	
WELL_ACTIVITY_SOURCE	VARCHAR2 (20)	Null	-	
WELL_ACTIVITY_TYPE	VARCHAR2 (20)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

19. Tabel PPDM38_VT_PROD_STRING

- Nama : PPDM38_VT_PROD_STRING
- Deskripsi isi : Berisi string dari semua data *well*
- *Primary key* : *Uwi, string_id, string_code*
- *Foreign key* : *Uwi*

Tabel B. 19 Basis data PPDM38 vt prod string

<i>Nama Field</i>	<i>Type dan Length</i>	<i>Null</i>	<i>Default</i>	<i>Ket</i>
UWI*/**	VARCHAR2 (20)	Not null	-	
STRING_ID*	VARCHAR2 (20)	Not null	-	
STRING_CODE*	VARCHAR2 (30)	Not null	-	
STRING_NAME	VARCHAR2 (30)	Null	-	
ALT_NAME	VARCHAR2 (30)	Null	-	
PPDM_STRING_CODE	VARCHAR2 (30)	Null	-	
STATUS_TYPE	VARCHAR2 (20)	Null	-	
CURRENT_STATUS	VARCHAR2 (20)	Null	-	
CURRENT_STATUS_DATE	DATE	Null	-	
PROD_STRING_TYPE	VARCHAR2 (20)	Null	-	
PROFILE_TYPE	VARCHAR2 (20)	Null	-	
EFFECTIVE_DATE	DATE	Null	-	
EXPIRY_DATE	DATE	Null	-	
OP_PRODUCTIONUNIT_CODE	VARCHAR2 (255)	Null	-	
P_SUB_PRODUCTIONUNIT_C	VARCHAR2 (255)	Null	-	
LICENCE	VARCHAR2 (255)	Null	-	
OP_AREA_CODE	VARCHAR2 (255)	Null	-	
OP_SUB_AREA_CODE	VARCHAR2 (255)	Null	-	
ASSET_AREA	VARCHAR2 (255)	Null	-	
FIELD	VARCHAR2 (255)	Null	-	
AREA	VARCHAR2 (255)	Null	-	
ARSE	VARCHAR2 (255)	Null	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	Null	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	Null	-	
ON_PRODUCTION_DATE	DATE	Null	-	
PRODUCTION_METHOD	VARCHAR2 (20)	Null	-	
WELL_NAME	VARCHAR2 (255)	Null	-	
TIMESTAMP	DATE	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

20. Tabel KLO_ASSET_DOC

- Nama : KLO_ASSET_DOC
- Deskripsi isi : Berisi dokumen *well*
- *Primary key* : *Wellnm*
- *Foreign key* : *Wellnm*

Tabel B. 20 Basis data KLO asset doc

Nama Field	Type dan Length	Null	Default	Ket
WELLNM**/*	VARCHAR2 (30)	Not null	-	
FOPENPERF	VARCHAR2 (500)	Null	-	
FCURRCOMPL	VARCHAR2 (500)	Null	-	
FLOG1	VARCHAR2 (500)	Null	-	
FLOG2	VARCHAR2 (500)	Null	-	
FLOG3	VARCHAR2 (500)	Null	-	
FMAP1	VARCHAR2 (500)	Null	-	
FMAP2	VARCHAR2 (500)	Null	-	
WELLORDER	VARCHAR2 (500)	Null	-	
XSEC1	VARCHAR2 (500)	Null	-	
XSEC2	VARCHAR2 (500)	Null	-	
XSEC3	VARCHAR2 (500)	Null	-	
LABEL	VARCHAR2 (500)	Null	-	
PARAFFIN	VARCHAR2 (500)	Null	-	
PATHFOPENPERF	VARCHAR2 (500)	Null	-	
PATHFCURRCOMPL	VARCHAR2 (500)	Null	-	
PATHFLOG1	VARCHAR2 (500)	Null	-	
PATHFLOG2	VARCHAR2 (500)	Null	-	
PATHFLOG3	VARCHAR2 (500)	Null	-	
PATHFMAP1	VARCHAR2 (500)	Null	-	
PATHFMAP2	VARCHAR2 (500)	Null	-	
PATHXSEC1	VARCHAR2 (500)	Null	-	
PATHXSEC2	VARCHAR2 (500)	Null	-	
PATHXSEC3	VARCHAR2 (500)	Null	-	
PATHPARAFFIN	VARCHAR2 (500)	Null	-	
ROW_CHANGED_BY	VARCHAR2 (30)	Null	-	
ROW_CHANGED_DATE	DATE	Null	-	
ROW_CREATED_BY	VARCHAR2 (30)	Null	-	
ROW_CREATED_DATE	DATE	Null	-	
ROW_QUALITY	VARCHAR2 (20)	Null	-	
FSURVEILLANCE	VARCHAR2 (500)	Null	-	
PATHSURVEILLANCE	VARCHAR2 (500)	Null	-	
PATHSLICKLINE	VARCHAR2 (500)	Null	-	
FSLICKLINE	VARCHAR2 (500)	Null	-	
PATHWCAP	VARCHAR2 (500)	Null	-	
FWCAP	VARCHAR2 (500)	Null	-	
PATHWSTATUS	VARCHAR2 (500)	Null	-	
FWSTATUS	VARCHAR2 (500)	Null	-	
WELLNM	VARCHAR2 (30)	Null	-	
FOPENPERF	VARCHAR2 (500)	Null	-	

FCURRCOMPL	VARCHAR2 (500)	Null	-	
FLOG1	VARCHAR2 (500)	Null	-	
FLOG2	VARCHAR2 (500)	Null	-	
FLOG3	VARCHAR2 (500)	Null	-	
FMAP1	VARCHAR2 (500)	Null	-	
FMAP2	VARCHAR2 (500)	Null	-	
WELLORDER	VARCHAR2 (500)	Null	-	
XSEC1	VARCHAR2 (500)	Null	-	
XSEC2	VARCHAR2 (500)	Null	-	
XSEC3	VARCHAR2 (500)	Null	-	
LABEL	VARCHAR2 (500)	Null	-	
PARAFFIN	VARCHAR2 (500)	Null	-	
PATHFOPENPERF	VARCHAR2 (500)	Null	-	
PATHFCURRCOMPL	VARCHAR2 (500)	Null	-	
PATHFLOG1	VARCHAR2 (500)	Null	-	
PATHFLOG2	VARCHAR2 (500)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

21. Tabel KLO_ASSET_COM

- Nama : KLO_ASSET_COM
- Deskripsi isi : Berisi data *comment (remark)*
- *Primary key* : *Wellnm, comment_by, comment_date*
- *Foreign key* : *Wellnm*

Tabel B. 21 Basis data KLO asset com

Nama Field	Type dan Length	Null	Default	Ket
WELLMN*/**	VARCHAR2 (30)	Not null	-	
COMMENT_BY*	VARCHAR2 (30)	Not null	-	
COMMENT_DATE*	DATE	Not null	-	
COMMENT_TEXT	VARCHAR2 (1000)	Null	-	
LAST_COMMENT	VARCHAR2 (1)	Null	-	

Sumber : PT.Chevron Pacific Indonesia (2013)

Keterangan : * = *Primary Key*

** = *Foreign Key*

B.2. Perancangan View (Lanjutan)

1. V_DSC_LPO_SUMMARY_WO_SHUTIN

Berikut ini merupakan *query view* pada V_DSC_LPO_SUMMARY_WO_SHUTIN.

```
Create or replace view v_dsc_lpo_summary_wo_shutin as
Select 'KLO' area, a.today_prod, a.today_lpo, a.ytd_prod,
a.ytd_lpo, (a.today_prod/(a.today_prod + lpo.today_
eventcat_lpo)) * 100 today_prod_eff, (a.ytd_prod/(a.ytd_prod
+ lpo.ytd_eventcat_lpo)) * 100 ytd_prod_eff
From (select sum (sub_area_lpo.prod) today_prod, sum
(sub_area_lpo.lpo) today_lpo, sum (ytd_sub_area_lpo.prod)
ytd_prod, sum (ytd_sub_area_lpo.lpo) ytd_lpo
From (select status_date, sub_area, prod, lpo
From (select status_date, 'attaka' sub_area, attaka
prod, attaka_lpo lpo
From KLO.v_dsc_today_lpo_summary_area union
select status_date, 'NIB' SUB_AREA, nib prod, nib_lpo
lpo
from KLO.v_dsc_today_lpo_summary_area
union
select status_date, 'sepinggan' sub_area, sep prod,
sep_lpo lpo
from KLO.v_dsc_today_lpo_summary_area
(select status_date, sub_area, prod, lpo
from (select status_date, 'attaka' sub_area, attaka
prod, attaka_lpo lpo
from KLO.v_dsc_ytd_lpo_summary_area
union
select status_date, 'nib' sub_area, nib prod,
nib_lpo lpo
from KLO.v_dsc_ytd_lpo_summary_area
= ytd_sub_area_lpo.status_date
and sub_area_lpo.sub_area = ytd_sub_area_lpo.sub_area) a,
(select sum(well_lpo_control) today_eventcat_lpo,
sum(ytd_well_lpo_control) ytd_eventcat_lpo from
v_dsc_lpo_eventcat_wo_shutin) lpo
```

Dari *query view* pada V_DSC_LPO_SUMMARY_WO_SHUTIN diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 22 View v dsc lpo summary wo shutin

Nama Field	Type dan Length	Default	Ket
AREA	VARCHAR2 (32)	-	
TODAY_PROD	NUMBER	-	
TODAY_LPO	NUMBER	-	
YTD_PROD	NUMBER	-	
YTD_LPO	NUMBER	-	
YTD_KLO_LPO_UNCONTROL	NUMBER	-	
YTD_KLO_LPO_CONTROL	NUMBER	-	
TODAY_PROD_EFF	NUMBER	-	
YTD_PROD_EFF	NUMBER	-	

2. V_DSC_LPO_EVENTCAT_WO_SHUTIN

Berikut ini merupakan *query view* pada V_DSC_LPO_EVENTCAT_WO_SHUTIN.

```
Create or replace view v_dsc_lpo_eventcat_wo_shutin as
Select ld.subarea, sum(ytd_cl.lpo) ytd_well_lpo_control,
sum(ytd_ucl.lpo) ytd_well_lpo_uncontrol, sum(mtd_cl.lpo)
mtd_well_lpo_control, sum(mtd_ucl.lpo) mtd_well_lpo_
uncontrol, sum(today_cl.lpo) well_lpo_control,
From (select distinct decode(y.field, attaka, attaka, nib,
serang, nib, santan, nib, sapi, nib, sepinggan, sepinggan,
sedandang, sepinggan, seguni, sepinggan, sejadi,
sepinggan, yakin, yakin, bangkirai, yakin, mahoni, yakin,
pantai, yakin, seturian, yakin, west_seno, ws_fpu) subarea
from well_last_data y) ld,
(select decode(y.field, attaka, attaka, melahin, nib,
kerindungan, nib, serangan, nib, santan, nib, sapi, nib,
sepinggan, sepinggan, sedandang, sepinggan, seguni,
sepinggan, sejadi, sepinggan, yakin, yakin, bangkirai,
yakin, mahoni, yakin, pantai, yakin, seturian, yakin,
west_seno, ws_fpu) subarea,
Sum(nvl(y.defered_oil_prod, 0) + nvl(y.defered_gas_prod
/ 6, 0)) lpo
From dsc_lpo_day_wo_shutin y
Where y.event_cat_ref_type = 'controllable'
And y.production_day = trunc(sysdate - 1)
Group by decode(y.field, attaka, attaka, melahin,
nib, kerindungan, nib, serangan, nib, santan, nib,
sapi, nib, sepinggan, sepinggan, sedandang,
sepinggan, seguni, sepinggan, sejadi, sepinggan,
yakin, yakin, bangkirai, yakin, mahoni, yakin,
pantai, yakin, seturian, yakin, west_seno, ws_fpu))
today_cl, (select decode(y.field, attaka, attaka,
melahin, nib, sepinggan,sejadi, sepinggan, yakin,
yakin, bangkirai, yakin, mahoni, yakin, pantai,
yakin, seturian, yakin, west_seno, ws_fpu) subarea,
Sum(nvl(y.defered_oil_prod, 0) +
nvl(y.defered_gas_prod / 6, 0)) lpo
from dsc_lpo_day_wo_shutin y
where y.event_cat_ref_type = 'uncontrollable'
and y.production_day between trunc(sysdate, 'mon')
and trunc(sysdate - 1)
group by decode(y.field, attaka, attaka, melahin,
nib, kerindungan, nib, serangan, nib, santan, nib,
sapi, nib, sepinggan, sepinggan, sedandang,
sepinggan, seguni, sepinggan, sejadi, sepinggan,
yakin, yakin, bangkirai, yakin, mahoni, yakin,
pantai, yakin, seturian, yakin, west_seno,
ws_fpu)) mtd_ucl,
where ld.subarea = ytd_cl.subarea(+)
and ld.subarea = ytd_ucl.subarea(+)
and ld.subarea = mtd_cl.subarea(+)
and ld.subarea = mtd_ucl.subarea(+)
and ld.subarea = today_cl.subarea(+)
and ld.subarea = today_ucl.subarea(+)
group by ld.subarea
```

Dari *query view* pada V_DSC_LPO_EVENTCAT_WO_SHUTIN diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 23 View v dsc lpo eventcat wo shutin

Nama Field	Type dan Length	Default	Ket
SUBAREA	VARCHAR2 (9)	-	
YTD_WELL_LPO_CONTROL	NUMBER	-	
YTD_WELL_LPO_UNCONTROL	NUMBER	-	
MTD_WELL_LPO_CONTROL	NUMBER	-	
MTD_WELL_LPO_UNCONTROL	NUMBER	-	
WELL_LPO_CONTROL	NUMBER	-	
WELL_LPO_UNCONTROL	NUMBER	-	

3. V_DSC_PRODEFF_FIELD_WO_SHUTIN

Berikut ini merupakan *query view* pada V_DSC_PRODEFF_FIELD_WO_SHUTIN.

```
create or replace view v_dsc_prodeff_field_wo_shutin as
select  a.daytime,  a.field,  a.prod,  cl.lpo_control  +
ucl.lpo_uncontrol lpo,
        coalesce(round(decode(a.prod + cl.lpo_control,
0,
0,
(a.prod / (a.prod + cl.lpo_control)) * 100),
2),
100) as prod_eff,
        cl.lpo_control,  ucl.lpo_uncontrol,  ytd_cl.lpo  ytd_lpo_
control,  ytd_ucl.lpo  ytd_lpo_uncontrol,  ytd_prod.prod
ytd_prod,
        coalesce(round(decode(ytd_prod.prod + ytd_cl.lpo,
0,
0,
(ytd_prod.prod / (ytd_prod.prod + ytd_cl.lpo)) * 100),
2),
100) as ytd_prod_eff
from (select trunc(sysdate - 1) daytime, field, sum(boepd) prod
      from v_dsc_well_except_efficiency
      group by field, trunc(sysdate - 1)) a,
(select trunc(sysdate - 1) daytime, field,
        sum(nvl(a.defered_oil_prod,0)
+ (nvl(a.defered_gas_prod,0)/6))
        lpo_control
      from dsc_lpo_day_wo_shutin a
      where a.event_cat_ref_type = 'controllable'
            and production_day = trunc(sysdate - 1)
            group by field, trunc(sysdate - 1)) cl,
where a.field = cl.field(+)    and a.field = ucl.field(+)
      and a.field = ytd_cl.field(+)
      and a.field = ytd_ucl.field(+)
      and a.field = ytd_prod.field(+)
```


Dari *query view* pada V_DSC_PRODEFF_FIELD_WO_SHUTIN diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 24 View v dsc prodeff field wo shutin

Nama Field	Type dan Length	Default	Ket
DAYTIME	DATE	-	
FIELD	VARCHAR2 (32)	-	
PROD	VARCHAR2 (32)	-	
LPO	VARCHAR2 (32)	-	
PROD_EFF	NUMBER	-	
LPO_CONTROL	NUMBER	-	
LPO_UNCONTROL	NUMBER	-	
YTD_LPO_CONTROL	NUMBER	-	
YTD_LPO_UNCONTROL	NUMBER	-	
YTD_PROD	NUMBER	-	
YTD_PROD_EFF	NUMBER	-	

4. V_DSC_LPO_AREA_WO_SHUTIN

Berikut ini merupakan *query view* pada V_DSC_LPO_AREA_WO_SHUTIN.

```
create or replace view v_dsc_lpo_area_wo_shutin as
select      ld.area,          sum(ytd_cl.lpo)      ytd_well_lpo_control,
            sum(ytd_ucl.lpo) ytd_well_lpo_uncontrol,
            sum(mtd_cl.lpo)   mtd_well_lpo_control,      sum(mtd_ucl.lpo)
            mtd_well_lpo_      uncontrol,          sum(today_cl.lpo)
            well_lpo_control, sum(today_ucl.lpo) well_lpo_uncontrol
from (select distinct op_area_code area
      from well_last_data y
      where op_area_code <> 'plgas_operation') ld,
      (select op_area_code area,
            sum(nvl(y.defered_oil_prod, 0) + nvl(y.defered_gas_prod /
            6, 0)) lpo
      from dsc_lpo_day_wo_shutin y
      where y.event_cat_ref_type = 'controllable'
            and y.production_day = trunc(sysdate - 1)
            and op_area_code <> 'plgas_operation'
            group by op_area_code) today_cl,
      (select op_area_code area,
            sum(nvl(y.defered_oil_prod, 0) + nvl(y.defered_gas_prod /
            6, 0)) from dsc_lpo_day_wo_shutin y
      where y.event_cat_ref_type = 'uncontrollable'
            and y.production_day between trunc(sysdate, 'year') and
            trunc(sysdate - 1)
            and op_area_code <> 'plgas_operation'
            group by op_area_code) ytd_ucl
where ld.area = ytd_cl.area(+)
      and ld.area = ytd_ucl.area(+) and ld.area =
      mtd_cl.area(+)and ld.area = mtd_ucl.area(+)and ld.area =
      today_cl.area(+)and ld.area = today_ucl.area(+)
group by ld.area
```

Dari *query view* pada V_DSC_LPO_AREA_WO_SHUTIN diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 25 View v dsc lpo area wo shutin

Nama Field	Type dan Length	Default	Ket
AREA	VARCHAR2 (32)	-	
YTD_WELL_LPO_CONTROL	NUMBER	-	
YTD_WELL_LPO_UNCONTROL	NUMBER	-	
MTD_WELL_LPO_CONTROL	NUMBER	-	
MTD_WELL_LPO_UNCONTROL	NUMBER	-	
WELL_LPO_CONTROL	NUMBER	-	
WELL_LPO_UNCONTROL	NUMBER	-	

5. KLO_V_DSC_WELL_EXCEPTION

Berikut ini merupakan *query view* pada KLO_V_DSC_WELL_EXCEPTION.

```
Create or replace view v_dsc_well_exception as
Select z.op_area_code area,
      Decode(z.field,   attaka,   attaka,   melahin,   nib,
kerindingan, nib,   serang, nib,   santan, nib,   sapi, nib,
sepinggan,   sepinggan,   sedandang,   sepinggan,   seguni,
sepinggan,   sejadi,   sepinggan,   yakin,   yakin,   bangkirai,
yakin,   mahoni,   yakin,   pantai,   yakin,   seturian,   yakin,
west_seno,   ws_fpu) sub_area,
      Substr(z.op_fcty_1_code,5,length(z.op_fcty_1_code)-3)
platform , z.field field, x.string_name, x.string_code,
      max(z.fm_gas ) mscfd, max(z.water_rate) bwpd,
      Max(z.gross_fluid) bfpd, max(z.oil_rate) bopd, max(t.slope)
slope, max(t.intercept) intercept,
      To_char(max(z.test_date),'dd-mon-yyyy')      last_test_date,
      'declining' condition,
      To_char(min(t.min_test_date),'dd-mon-yyyy') min
      _test_date, max(z.water_cut) bsw, max(z.gor) gor,
      Max (z.glor)   glor,max (z.wh_csg_press)   csg_press,max
      (z.wh_press) thp,max (z.wh_temp) wh_temp,
      max(z.gl_choke) gl_choke,max(z.surf_choke) surf_choke
From KLO.v_dsc_well_test_linreg_eq t, KLO.v_last_well_test x,
KLO.v_last_well_test z
Where x.string_code = t.string_code and t.slope < 0
And x.dense_rank < 6
And z.dense_rank= 1
And z.string_code= x.string_code
Group by z.op_area_code,      substr(z.op_fcty_1_code, 5,
length(z.op_fcty_1_code)-3),
      Decode(z.field, sepinggan, sedandang, sepinggan, seguni,
sepinggan, sejadi, sepinggan, yakin, yakin, bangkirai,
yakin, mahoni, yakin, pantai, yakin, seturian, yakin,
west_seno, ws_fpu),      z.field,      x.string_name,      x.
String_codem,
```

Query view pada KLO_V_DSC_WELL_EXCEPTION (lanjutan)

```

union
select z.op_area_code area,
      decode(z.field, attaka, attaka, melahin, nib, kerindingan,
            nib,  serang, nib, santan, nib, sapi, nib, sepinggan,
            sepinggan, sedandang, sepinggan, seguni, sepinggan,
            sejadi, sepinggan, yakin, yakin, bangkirai, yakin,
            mahoni, yakin, pantai, yakin, seturian, yakin,
            west_seno, ws_fpu) sub_area,
      substr(z.op_fcty_1_code,5,length(z.op_fcty_1_code)-3)
      platform , z.field field, x.string_name, x.string_code,
      max(z.fm_gas ) mscfd, max(z.water_rate) bwpd,
      max(z.gross_fluid) bfpd, max(z.oil_rate) bopd, max(t.slope)
      slope, max(t.intercept) intercept,
      to_char(max(z.test_date),'dd-mon-yyyy')      last_test_date,
      'inclining' condition,
      to_char(min(t.min_test_date),'dd-mon-yyyy')   min_test_date,
      max(z.water_cut) bsw , max(z.gor) gor,
      max (z.glor) glor,max (z.wh_csg_press) csg_press,max
      (z.wh_press) thp,max (z.wh_temp) wh_temp,
      max(z.gl_choke) gl_choke,max(z.surf_choke) surf_choke
from KLO.v_dsc_well_test_linreg_eq t, KLO.v_last_well_test x,
KLO.v_last_well_test z
where x.string_code = t.string_code
and t.slope = 0
      and x.dense_rank < 6
      and z.dense_rank= 1
      and z.string_code= x.string_code
group by z.op_area_code,      substr(z.op_fcty_1_code, 5,
length(z.op_fcty_1_code)-3),
      decode(z.field, attaka, attaka, melahin, nib, kerindingan,
            nib,  serang, nib, santan, nib, sapi, nib, sepinggan,
            sepinggan, sedandang, sepinggan, seguni, sepinggan, sejadi,
            sepinggan, yakin, yakin, bangkirai, yakin, mahoni, yakin,
            pantai, yakin, seturian, yakin, west_seno, ws_fpu),
      z.field, x.string_name,

```

Dari *query view* pada KLO_V_DSC_WELL_EXCEPTION diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 26 View KLO v dsc well exception

Nama Field	Type dan Length	Default	Ket
AREA	VARCHAR2 (32)	-	
SUB_AREA	VARCHAR2 (9)	-	
PLATFORM	VARCHAR (112)	-	
FIELD	VARCHAR2 (32)	-	
STRING_NAME	VARCHAR2 (30)	-	
STRING_CODE	VARCHAR2 (30)	-	
MSCFD	NUMBER	-	
BWPD	NUMBER	-	
BFPD	NUMBER	-	
BOPD	NUMBER	-	
SLOPE	NUMBER	-	
INTERCEPT	NUMBER	-	

LAST_TEST_DATE	DATE	-	
CONDITION	VARCHAR2 (9)	-	
MIN_TEST_DATE	DATE	-	
BSW	NUMBER	-	
GOR	NUMBER	-	
GLOR	NUMBER	-	
CSG_PRESS	NUMBER	-	
THP	NUMBER	-	
WH_TEMP	NUMBER	-	
BOEPD	NUMBER	-	
GLR	NUMBER	-	
WOR	NUMBER	-	
GL_RATE	NUMBER	-	
GL_CHOKE	NUMBER	-	
SURF_CHOKE	NUMBER	-	

6. KLO_V_DSC_TODAY_LPO_SUMMARY_AREA

Berikut ini merupakan *query view* pada KLO_V_DSC_TODAY_LPO_SUMMARY_AREA.

```
Create or replace view v_dsc_today_lpo_summary_area as
Select trunc(sysdate-1) status_date, attaka.prod attaka,
nib.prod nib,sep.prod sep, yakin.prod yakin, wss.prod wss,
attaka_lpo.lpo attaka_lpo,nib_lpo.lpo nib_lpo, sep_lpo.lpo
sep_lpo,yakin_lpo.lpo yakin_lpo,wss_lpo.lpo wss_lpo
From ( Select
(select nvl(sum(flow_rate) / 6, 0)
From PPDM38.pden_flow_measurement t
Where pden_id in ('ca_fe_1103')
And measurement_date between trunc(sysdate-1)
+
(select nvl(sum((1 - bs_w) * t.production_volume), 0)
From PPDM38.pden_flow_measurement t
Where pden_id in ('atta_santa_oil_corfel09')
and measurement_date between trunc(sysdate-1)
) prod
From dual) attaka,
(select(select nvl(sum(case when pden_id in
(sera_liftx_gas_fe1400, sera_fuelx_gas_fe1912,
mela_fuelc_gas_tot_fitfltcl00,keri_hpxxx_gas_fuel_f
Tv2) then -flow_rate else flow_rate end) / 6, 0)
From PPDM38.pden_flow_measurement t
Where pden_id in (ca_fe_102, sera_liftx_gas_fe1400,
sera_xxxxx_gas_fe4100, sera_fuelx_gas_fe1912,
sera_hpxxx_gas_fe1402, mela_fuelc_gas_tot_fitfltcl00,
ca_kerindingan_gas_aloc, keri_hpxxx_gas_fuel_ftv2,
sapi_exp_lng_m850, sapi_exp_lng_m851, sapi_flare_ft406,
sapi_fuel_fqt310, mk_lp_gas)
and measurement_date between trunc(sysdate-1) and
trunc(sysdate-1))
+
(select nvl(sum((1 - bs_w) * t.production_volume), 0)
```

Query view pada KLO_V_DSC_TODAY_LPO_SUMMARY_AREA (lanjutan)

```

from PPDM38.pden_flow_measurement t
where pden_id in ('seno_fpu_oil_mtr_santan_ft1251')
      and measurement_date between trunc(sysdate-1) and
      trunc(sysdate-1)
) prod
from dual) wss,
(select
  sum(nvl(a.defered_oil_prod_corrected,0)+(nvl(a.defered_gas
    _prod,0)/6)) lpo
from PPDM38.v_well_deferment_daily a
where potential_method <> 'na'
      and trunc(start_date_event) between trunc(sysdate-1)
      and trunc(sysdate-1)
      and field in ('attaka')) attaka_lpo,(
select sum(nvl(a.defered_oil_prod_corrected, 0) +
  (nvl(a.defered_gas_prod, 0) / 6)) lpo
from PPDM38.v_well_deferment_daily a
where potential_method <> 'na'
      and trunc(start_date_event) between trunc(sysdate-1)
      and trunc(sysdate-1)
      and field in ('west_seno')) wss_lpo

```

Dari query view pada KLO_V_DSC_TODAY_LPO_SUMMARY_AREA diatas, menghasilkan struktur view seperti dibawah ini.

Tabel B. 27 View KLO v dsc today lpo summary area

Nama Field	Type dan Length	Default	Ket
STATUS_DATE	DATE	-	
ATTAKA	NUMBER	-	
NIB	NUMBER	-	
SEP	NUMBER	-	
YAKIN	NUMBER	-	
WSS	NUMBER	-	
ATTAKA_LPO	NUMBER	-	
NIB_LPO	NUMBER	-	
SEP_LPO	NUMBER	-	
YAKIN_LPO	NUMBER	-	
WSS_LPO	NUMBER	-	

7. KLO_V_DSC_YTD_LPO_SUMMARY_AREA

Berikut ini merupakan *query view* pada KLO_V_DSC_YTD_LPO_SUMMARY_AREA.

```
Create or replace view v_dsc_ytd_lpo_summary_area as
Select trunc(sysdate-1) status_date, attaka.prod attaka,
      nib.prod nib, sep.prod sep, yakin.prod yakin, wss.prod wss,
      attaka_lpo.lpo attaka_lpo, nib_lpo.lpo nib_lpo, sep_lpo.lpo
      sep_lpo, yakin_lpo.lpo yakin_lpo, wss_lpo.lpo wss_lpo
From (
Select
(select nvl(sum(flow_rate) / 6, 0)
From PPDM38.pden_flow_measurement t
Where pden_id in ('ca_fe_1103')
      and measurement_date between trunc(sysdate, 'year') and
      trunc(sysdate-1))
+
(select nvl(sum((1 - bs_w) * t.production_volume), 0)
From PPDM38.pden_flow_measurement t
Where pden_id in (atta_santa_oil_corfel09)
      and measurement_date between trunc(sysdate, year) and
      trunc(sysdate-1)
) prod
From dual) attaka,
(select
(select nvl(sum(case when pden_id in (sera_liftx_gas_fel400,
      sera_fuelx_gas_fel912, mela_fuelc_gas_tot_fitfltc100,
      kerihpxxx_gas_fuel_ftv2) then -flow_rate else flow_rate
end) / 6, 0)
From PPDM38.pden_flow_measurement t
Where pden_id in (ca_fe_102, sera_liftx_gas_fel400,
      sera_XXXXX_gas_fel4100, sera_fuelx_gas_fel912,
      sera_hpXXX_gas_fel402, mela_fuelc_gas_tot_fitfltc100,
      ca_kerindungan_gas_aloc, kerihpXXX_gas_fuel_ftv2,
      sapi_exp_lng_m850, sapi_exp_lng_m851,)
      and measurement_date between trunc(sysdate, year) and
      trunc(sysdate-1)) +
select nvl(sum((1 - bs_w) * t.production_volume), 0)
from PPDM38.pden_flow_measurement t
where pden_id in ('seno_fpu_oil_mtr_santan_ft1251')
      and measurement_date between trunc(sysdate, 'year') and
      trunc(sysdate-1)
) prod
from dual) wss, (
select sum(nvl(a.deferred_oil_prod_corrected, 0) + (nvl
(a.deferred_gas_prod, 0) / 6)) lpo
from PPDM38.v_well_deferment_daily a
where potential_method <> 'na'
      and trunc(start_date_event) between trunc(sysdate, 'year')
      and trunc(sysdate-1)
      and field in ('west_seno')) wss_lpo
```

Dari *query view* pada KLO_V_DSC_YTD_LPO_SUMMARY_AREA diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 28 View KLO v dsc ytd lpo summary area

Nama Field	Type dan Length	Default	Ket
STATUS_DATE	DATE	-	
ATTAKA	NUMBER	-	
NIB	NUMBER	-	
SEP	NUMBER	-	
YAKIN	NUMBER	-	
WSS	NUMBER	-	
ATTAKA_LPO	NUMBER	-	
NIB_LPO	NUMBER	-	
SEP_LPO	NUMBER	-	
YAKIN_LPO	NUMBER	-	
WSS_LPO	NUMBER	-	

8. KLO_V_DSC_WELL_TEST_LINREG_EQ

Berikut ini merupakan *query view* pada KLO_V_DSC_WELL_TEST_LINREG_EQ.

```
create or replace view v_dsc_well_test_linreg_eq as
select string_name,string_code,
       case when (n*sigma_x_sqr - sqr_sigma_x) = 0 then
         0 else
         (n*sigma_xy - sigma_x*sigma_y)/(n*sigma_x_sqr - sqr_sigma_x)
       end slope,
       case when (n*sigma_x_sqr - sqr_sigma_x) = 0 then
         0 else
         (sigma_y - (n*sigma_xy - sigma_x*sigma_y)/(n*sigma_x_sqr -
         sqr_sigma_x)*sigma_x)/n
       end intercept,
       n, avg_y, min_test_date, max_test_date
from (select a.string_name, string_code, sum(a.test_date -
to_date('01-jan-1900')+2) sigma_x,
       sum((a.total_gas/6)+ a.oil_rate) sigma_y,
       count(a.string_name) n,
       sum((a.test_date - to_date('01-jan-1900')+2) *
((a.total_gas/6)+ a.oil_rate)) sigma_xy,
       sum((a.test_date - to_date('01-jan-1900')+2) * (a.test_date
- to_date('01-jan-1900')+2)) sigma_x_sqr,
       (sum (a.test_date - to_date ('01-jan-1900') + 2) * sum
(a.test_date - to_date ('01-jan-1900') + 2)) sqr_sigma_x,
       avg((a.total_gas/6)+ a.oil_rate) avg_y,
       min(a.test_date) min_test_date, max(a.test_date)
max_test_date
from select x.dense_rank rank, x.string_name ,x.string_code,
nvl(x.fm_gas,0) total_gas, x.test_date test_date,
nvl(x.oil_rate,0) oil_rate
from KLO.v_last_well_test x
where x.dense_rank < 6 ) a group by a.string_name,
string_code )
```

Dari *query view* pada KLO_V_DSC_WELL_TEST_LINREG_EQ diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 29 View KLO v dsc well test linreg eq

Nama Field	Type dan Length	Default	Ket
STRING_NAME	VARCHAR2 (30)	-	
STRING_CODE	VARCHAR2 (30)	-	
SLOPE	NUMBER	-	
INTERCEPT	NUMBER	-	
N	NUMBER	-	
AVG_Y	NUMBER	-	
MIN_TEST_DATE	DATE	-	
MAX_TEST_DATE	DATE	-	

9. KLO_V_LAST_WELL_TEST

Berikut ini merupakan *query view* pada KLO_V_LAST_WELL_TEST.

```
create or replace view v_last_well_test as
select  string_code,  string_name,  test_date,  bsw_percent,
        water_cut,  wh_csg_press,  wh_press,  wh_temp,  oil_rate,
        water_rate,  gross_fluid,  fm_gas,  total_gas,  gor,  glor,  glr,
        platform,
        op_sub_productionunit_code,  op_area_code,
        op_sub_area_code,  op_fcty_1_code,  op_fcty_2_code,  field,
        gl_rate,  gl_choke,  surf_choke,  sep_press,  wor,  dense_rank
from    (select  t.string_code,  t.string_name,  t.test_date,
        t.bsw_percent,  t.water_cut,  t.wh_csg_press,  t.wh_press,
        t.wh_temp,  t.oil_rate,  t.water_rate,  t.gross_fluid,
        t.fm_gas,  t.total_gas,  t.gor,  t.glor,  t.glr,  t.platform,
        t.op_sub_productionunit_code,  t.op_area_code,
        t.op_sub_area_code,  t.op_fcty_1_code,  t.op_fcty_2_code,
        field,  t.gl_rate,  t.gl_choke,  t.surf_choke,  t.sep_press,
        decode(t.gross_fluid,0,0,t.water_rate/t.gross_fluid) wor,
        dense_rank() over(partition by string_code order by
        test_date desc, test_time desc, test_num desc) dense_rank
from KLO.well_test t
where t.test_result_code = 'accepted'
      and (t.oil_rate > 0 or t.total_gas >0)
      ) x
where x.dense_rank <=10
```


Dari *query view* pada KLO_V_LAST_WELL_TEST diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 30 View KLO v last well test

<i>Nama Field</i>	<i>Type dan Length</i>	<i>Default</i>	<i>Ket</i>
STRING_CODE	VARCHAR2 (30)	-	
STRING_NAME	VARCHAR2 (30)	-	
TEST_DATE	DATE	-	
BSW_PERCENT	NUMBER	-	
WATER_CUT	NUMBER	-	
WH_CSG_PRESS	NUMBER	-	
WH_PRESS	NUMBER	-	
WH_TEMP	NUMBER	-	
OIL_RATE	NUMBER	-	
WATER_RATE	NUMBER	-	
GROSS_FLUID	NUMBER	-	
FM_GAS	NUMBER	-	
TOTAL_GAS	NUMBER	-	
GOR	NUMBER	-	
GLOR	NUMBER	-	
GLR	NUMBER	-	
PLATFORM	VARCHAR (112)	-	
OP_SUB_PRODUCTIONUNIT_CODE	VARCHAR2 (32)	-	
OP_AREA_CODE	VARCHAR2 (32)	-	
OP_SUB_AREA_CODE	VARCHAR2 (32)	-	
OP_FCTY_1_CODE	VARCHAR2 (32)	-	
OP_FCTY_2_CODE	VARCHAR2 (32)	-	
FIELD	VARCHAR2 (32)	-	
GL_RATE	NUMBER	-	
GL_CHOKE	NUMBER	-	
SURF_CHOKE	NUMBER	-	
SEP_PRESS	NUMBER	-	
WOR	NUMBER	-	
DENSE_RANK	NUMBER	-	

10. V_DSC_WELL_STATUS_DAY

Berikut ini merupakan *query view* pada V_DSC_WELL_STATUS_DAY.

```
Create or replace view v_dsc_well_status_day as
Select daytime, uwi, string_code, status
From (select td.daytime, ph.uwi, ph.string_code, ph.status_
      type, ph.status, ph.prod_string_type, dense_rank() over
      (partition by ph.string_code, daytime order by ph.start_time
      desc) dense_rank
From time_dim td
Left join string_status_hist ph
  On (td.daytime between trunk (ph.start_time) and trunk
      (nvl(ph.end_time, trunc(sysdate))))))
Where dense_rank = 1
And status in ('on', 'off')
and status_type = 'act'
```

Dari *query view* pada V_DSC_WELL_STATUS_DAY diatas, menghasilkan struktur *view* seperti dibawah ini.

Tabel B. 31 View v dsc well status day

Nama Field	Type dan Length	Default	Ket
DAYTIME	DATE	-	
UWI	VARCHAR (20)	-	
STRING_CODE	VARCHAR (30)	-	
STATUS	VARCHAR2 (20)	-	

DAFTAR RIWAYAT HIDUP



Penulis bernama Aditya Ayu Hapsari dilahirkan di Pekanbaru pada tanggal 20 Oktober 1991 dari ayahanda Kiesworo Hartono dan Ibunda Sumarti. Penulis anak ketiga dari 2 bersaudara yaitu Harma Didy Wicaksono,ST dan Tetra Hapsoro,ST.

Pada tahun 1995, Penulis masuk Taman Kanak-kanak (TK) Pertiwi. Kemudian tahun 1996 melanjutkan pendidikan ke Sekolah Dasar Pembangunan Bagan Batu dan tamat tahun 2003. Melanjutkan pendidikan di tingkat Sekolah Menengah Pertama (SMP) Negeri 14 Pekanbaru dan tamat pada tahun 2006. Setelah tamat SMP, Penulis melanjutkan pendidikan ke Sekolah Menengah Atas (SMA) Negeri 10 Pekanbaru dan tamat tahun 2009.

Setelah menamatkan pendidikan formal di tahun 2009, penulis masuk ke Universitas Islam Negeri (UIN) Sultan Syarif Kasim, sebagai mahasiswi Jurusan Teknik Informatika hingga menyelesaikan Laporan Tugas Akhir. Selama menjadi mahasiswi, Penulis pernah melaksanakan Kerja Praktek (KP) di Dinas Perkebunan Provinsi Riau tahun 2012.